Ingenieurfakultät Bau Geo Umwelt
Technische Universität München

Deep Vision in Optical Imagery: From Perception to Reasoning

Lichao Mou

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende:
Prof. Dr.-Ing. Liqiu Meng

Prüfende der Dissertation:

1. Prof. Dr.-Ing. habil. Xiaoxiang Zhu
2. Prof. Dr.-Ing. habil. Richard H. G. Bamler
3. Prof. Dr. Devis Tuia

Die Dissertation wurde am 19.12.2019 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 03.03.2020 angenommen.

Abstract

Deep learning has achieved extraordinary success in a wide range of tasks in computer vision field over the past years. Remote sensing data present different properties as compared to natural images/videos, due to their unique imaging technique, shooting angle, etc. For instance, hyperspectral images usually have hundreds of spectral bands, offering additional information, and the size of objects (e.g., vehicles) in remote sensing images is quite limited, which brings challenges for detection or segmentation tasks.

This thesis focuses on two kinds of remote sensing data, namely hyper/multi-spectral and high-resolution images, and explores several methods to try to find answers to the following questions:

- In comparison with natural images or videos in computer vision, the unique asset of hyper/multi-spectral data is their rich spectral information. But what this "additional" information brings for learning a network? And how do we take full advantage of these spectral bands?
- Remote sensing images at high resolution have pretty different characteristics, bringing challenges for several tasks, for example, small object segmentation. Can we devise tailored networks for such tasks?
- Deep networks have produced stunning results in a variety of perception tasks, e.g., image classification, object detection, and semantic segmentation. While the capacity to reason about relations over space is vital for intelligent species. Can a network/module with the capacity of reasoning benefit to parsing remote sensing data?

To this end, a couple of networks are devised to figure out what a network learns from hyperspectral images and how to efficiently use spectral bands. In addition, a multi-task learning network is investigated for the instance segmentation of vehicles from aerial images and videos. Finally, relational reasoning modules are designed to improve semantic segmentation of aerial images.

Zusammenfassung

Deep Learning hat in den letzten Jahren bei einer Vielzahl von Aufgaben im Bereich der Computer Vision außergewöhnliche Erfolge erzielt. Fernerkundungsdaten weisen aufgrund ihrer einzigartigen Bildgebungstechnik, ihres Aufnahmewinkels usw. verschiedene Eigenschaften im Vergleich zu natürlichen Bildern/Videos auf. Beispielsweise haben hyperspektrale Bilder in der Regel Hunderte von Spektralbändern, die zusätzliche Informationen bieten, und die Größe von Objekten (z.B. Fahrzeuge) in Fernerkundungsbildern ist sehr begrenzt, was Herausforderungen für Erkennungs- oder Segmentierungsaufgaben mit sich bringt.

Diese Arbeit konzentriert sich auf zwei Arten von Fernerkundungsdaten, nämlich hyperspektrale, multispektrale und hochauflösende Bilder, und untersucht mehrere Methoden, um zu versuchen, Antworten auf die folgenden Fragen zu finden:

- Im Vergleich zu natürlichen Bildern oder Videos im Computer-Vision-Bereich sind die einzigartigen Vorteile von Hyper/Multispektral-Daten ihre reichen Spektralinformationen. Aber was bringt diese "zusätzliche" Information für das Lernen eines Netzwerks? Und wie können wir die Vorteile dieser Spektralbänder voll ausschöpfen?

- Fernerkundungsbilder mit hoher Auflösung haben sehr unterschiedliche Eigenschaften und stellen eine Herausforderung für verschiedene Aufgaben dar, z.B. die Segmentierung kleiner Objekte. Können wir für solche Aufgaben maßgeschneiderte Netzwerke aufbauen?

- Tiefe Netzwerke haben bei einer Vielzahl von Wahrnehmungsaufgaben, wie z.B. Bildklassifizierung, Objekterkennung und semantische Segmentierung, erstaunliche Ergebnisse erbracht. Während die Fähigkeit, über Beziehungen über den Weltraum nachzudenken, für intelligente Arten lebenswichtig ist. Kann ein Netzwerk/Modul mit der Fähigkeit zum Schlussfolgern das Parsen von Fernerkundungsdaten unterstützen?

Zu diesem Zweck werden einige Netzwerke entwickelt, um herauszufinden, was ein Netzwerk aus hyperspektralen Bildern lernt und wie man Spektralbänder effizient nutzt. Darüber hinaus wird ein multitasking-fähiges Lernnetzwerk untersucht, zum Beispiel die Segmentierung von Fahrzeugen aus Luftbildern und Videos. Schließlich werden relationale Argumentationsmodule entwickelt, um die semantische Segmentierung von Luftbildern zu verbessern.

Acknowledgments

It is, in all probability, impossible to thank all people who gave me support to the completion of this doctoral thesis. Nevertheless, I take this opportunity trying to do so in what follows.

I would first like to thank my supervisor, Prof. Xiaoxiang Zhu, for the continuous support of my PhD study and related research, for her patience, enthusiasm, and excellent guide. I could not achieve all of this without her support.

I would also like to thank my thesis committee, Prof. Richard Bamler and Prof. Devis Tuia, for their kind support. Prof. Bamler's insights and enthusiasm on the theoretical understanding of deep learning often impress me, and discussions with him inspire me a lot. I am also amazed by works done by Prof. Tuia, from detecting mammals in UAV images to correcting building annotations, and his eye for deep learning in remote sensing inspires me in many ways.

Prof. Xuelong Li and Prof. Xiaoqiang Lu were my master thesis supervisors at CAS. I am very thankful to them as they opened the door of computer vision and machine learning for me at the very beginning.

I am so lucky to know the great research fellows in SiPEO-TUM and IMF-DLR. I would like to give special thanks to my family and my wife. Thank you, Mama, Papa, Fei!

The memory of the PhD life in Munich is a treasure of my life.

Lichao Mou
Munich, June 2019

Contents

1 Introduction

Nowadays we are in an era of big remote sensing data. Everyday huge volumes of spaceborne and airborne data are being produced by many different sensors, and these remotely sensed data can be used to understand how people and objects are organized in space. However, the increasingly availability of remote sensing data has raised a question: how to automatically and efficiently interpret these data?

1.1 Objectives

This thesis focuses on two types of optical data, namely hyper/multi-spectral imagery and high resolution remote sensing data, and aims to make use of deep networks to handle several tasks about these data. More specifically, the following problems are concerns of this thesis.

- **Problem 1: What do rich spectra of hyper/multi-spectral imagery bring for deep networks?**

Compared to natural images or videos in computer vision, the unique asset of hyper/multi-spectral data is their rich spectral information. Therefore, an intuitive question is what this "additional" information brings for applications using deep networks. And how do we take full advantage of these spectral bands?

- **Problem 2: Object detection and semantic segmentation in high resolution aerial images**

Remote sensing images at high resolution have pretty different characteristics in comparison with hyper/multi-spectral data and natural images, bringing challenges for several tasks, to name a few, vehicle detection and semantic segmentation in aerial images. Hence there is a need to devise tailored networks for these tasks.

- **Problem 3: Can remote sensing data analysis benefit from reasoning learning?**

Deep networks have produced stunning results in a variety of visual perception remote sensing tasks, e.g., image classification, scene recognition, object detection, and semantic segmentation. While the capacity to reason about relations among entities over space is vital for intelligent species. An interesting question is: can the interpretation of remote sensing data benefit from the idea of reasoning learning?

1.2 Thesis Organization

This is a cumulative dissertation where the above mentioned three problems are addressed in the following articles:

- Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639-3655, 2017.
- Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 391-406, 2018.
- Lichao Mou, Lorenzo Bruzzone, and Xiao Xiang Zhu. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 924-935, 2019.
- Lichao Mou and Xiao Xiang Zhu. Learning to pay attention on spectral domain: A spectral attention module-based convolutional network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 1, pp. 110-122, 2020.
- Lichao Mou and Xiao Xiang Zhu. Vehicle instance segmentation from aerial image and video using a multi-task learning residual fully convolutional network. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6699-6711, 2018.
- Lichao Mou, Yuansheng Hua, and Xiao Xiang Zhu. A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

2 A Glance at The Data

Before diving into methodologies, this chapter briefly introduces the data this thesis uses.

2.1 Hyper/Multi-Spectral Imagery and Challenges

A hyper/multi-spectral image is produced by an imaging equipment that is capable of capturing image data within particular wavelengths across the electromagnetic spectrum. A filter or an instrument that is sensitive to specific wavelengths can be used to separate different spectral channels. The wavelengths of a hyper/multi-spectral image usually expand the visible light range. Hence by using hyper/multi-spectral images, one can extract additional information that the human eye fails to capture with its receptors for red, green, and blue.

The main difference between hyperspectral data and multispectral data is the number of wavebands being imaged and how narrow the bands are. In general, multispectral images have 4 to 13 discrete border bands, while hyperspectral data usually consist of much narrower bands and have hundreds or even thousands of bands. Below is the information of several hyper/multi-spectral sensors/programs, which image the datasets used in the study of this thesis.

- ROSIS (Reflective Optics System Imaging Spectrometer): ROSIS was a compact airborne imaging spectrometer, which had been developed jointly by Dornier Satellite Systems (DSS, former MBB), GKSS Research Centre (Institute of Hydrophysics) and German Aerospace Center (DLR, Institute of Optoelectronics) based on an original design for a flight on ESA's EURECA platform. It is designed to cover a spectral range from 430 to 860 nm, and the number of bands is 115.
- AVIRIS (Airborne Visible/Infrared Imaging Spectrometer): AVIRIS is an imaging spectrometer instrument developed by the Jet Propulsion Laboratory (JPL) for Earth remote sensing. It delivers images in 224 contiguous spectral bands with wavelengths from 400 to 2500 nm.
- Landsat: Landsat is a long-running program for the acquisition of multispectral images of the Earth. Taking the Landsat-8 satellite as an example, it produces 11

> images with the following bands:
> Band 1: Coastal aerosol (0.43-0.45 µm)
> Band 2: Blue (0.45-0.51 µm)
> Band 3: Green (0.53-0.59 µm)
> Band 4: Red (0.64-0.67 µm)
> Band 5: Near infrared NIR (0.85-0.88 µm)
> Band 6: Short-wave Infrared SWIR 1 (1.57-1.65 µm)
> Band 7: Short-wave Infrared SWIR 2 (2.11-2.29 µm)
> Band 8: Panchromatic (0.50-0.68 µm)
> Band 9: Cirrus (1.36-1.38 µm)
> Band 10: Thermal Infrared TIRS 1 (10.60-11.19 µm)
> Band 11: Thermal Infrared TIRS 2 (11.50-12.51 µm)
>
> Each band has a spatial resolution of 30 m/pixel with the exception of bands 8, 10, and 11. Band 8 has a spatial resolution of 15 m/pixel. Band 10 and 11 have spatial resolutions of 100 m/pixel.

Although abundant spectral bands help in differentiating various materials, in pactice hyperspectral remote sensing images have intrinsic intra-class variations (samples in the same category may have different spectral signatures) and inter-class similarities (samples in different classes may share similar spectral signatures), which is an obstacle for the purpose of interpretation. Figure 2.1 shows average spectra of different classes on the Indian Pines agricultural site in northwestern Indiana, United States. It can be seen that average spectra of several categories are quite similar.

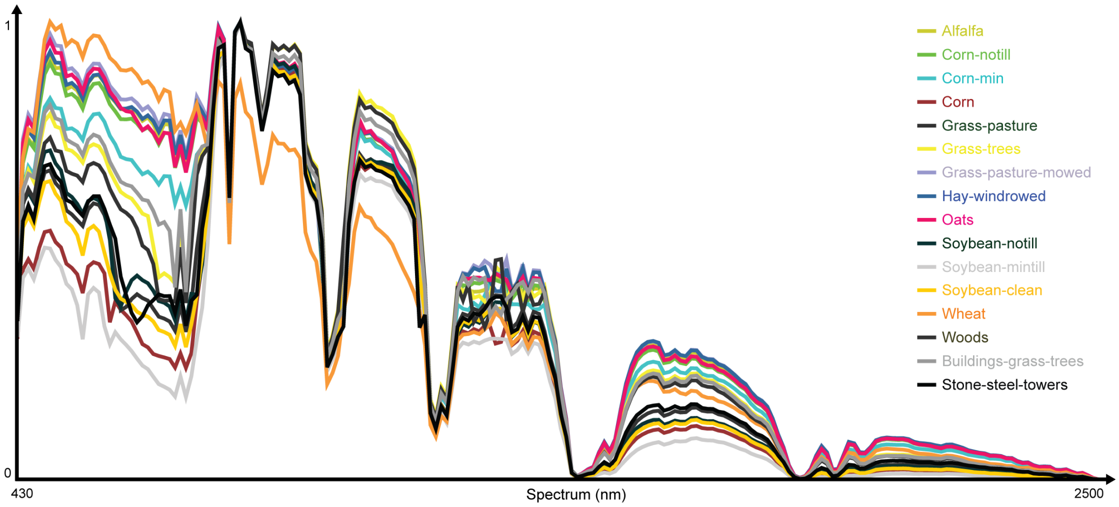


Figure 2.1: Illustration of inter-class similarities. [site: the Indian Pines agricultural site in northwestern Indiana, United States; sensor: AVIRIS.]

2.2 High Resolution Aerial Imagery and Challenges

In comparison with hyper/multi-spectral images, high resolution aerial images (GSD 5-30 cm) present quite different properties, bringing challenges for aerial image parsing, such as object detection and semantic segmentation. On the one hand, high resolution data deliver intricate spatial details (e.g., roof tiles, road markings, shadows, windows of vehicles, and branches of trees) emerge, which leads to big appearance differences within an object category. Figure 2.2 shows some examples of these challenges. One the other hand, available spectral information of such high resolution aerial images is less, as the spectral resolution of high spatial resolution sensors is usually limited to four (R-G-B-IR) or three bands (R-G-B).

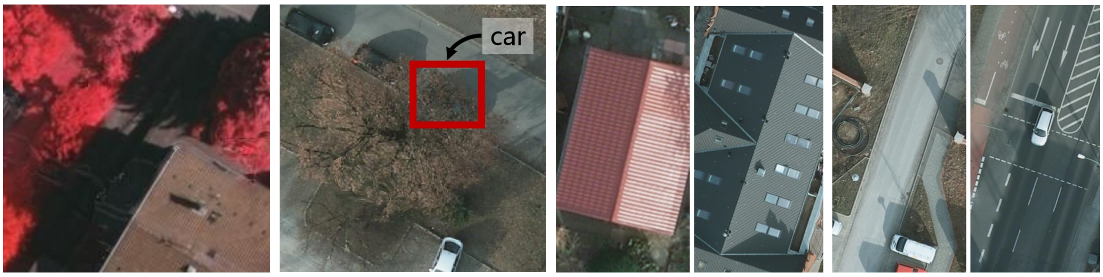


Figure 2.2: Illustration of challenges in high resolution aerial images for semantic segmentation tasks. From left to right: shadows, tree branches, big appearance variations within roofs, and roads.

3 State-of-the-Art

This chapter reviews existing works related to this thesis.

3.1 Deep Learning for Hyper/Multi-Spectral Image Analysis

Here several widely used networks and their applications in hyper/multi-spectral image analysis will be introduced.

3.1.1 Early Models

- **Models**

The first attempt in deep learning for hyper/multi-spectral data analysis is to make use of traditional autoencoder, Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN), and their variations. These models are both unsupervised and used to learn better feature representations than the raw data themselves. For instance, an autoencoder takes an input and first maps it to a latent feature space via a nonlinear encoder. Then the encoded features are used to reconstruct the input by a decoder, which is actually a reverse mapping. Finally, by reducing the Euclidean distance between the input and the reconstructed one, the autoencoder model can be trained. Unlike the autoencoder, an RBM is a generative stochastic model. The goal of an RBM is to recreate the input as accurately as possible. This is also the case for the autoencoder. During a forward pass, the input is modified by weights and biases and is used to activate the hidden layer. In the next pass, the activation from the hidden layer is modified by weights and biases and sent back to the input layer for activation. More specifically, the RBM model optimizes the following energy function:

$$\mathbb{E}(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{v} - \left(\boldsymbol{a}^T\boldsymbol{v} + \boldsymbol{b}^T\boldsymbol{h} + \boldsymbol{v}^T\boldsymbol{W}\boldsymbol{h}\right), \tag{3.1}$$

where $\boldsymbol{v}$ and $\boldsymbol{h}$ are visible and hidden units, respectively, $\boldsymbol{a}$ and $\boldsymbol{b}$ are the corresponding learnable bias weights, and $\boldsymbol{W}$ is a trainable weight matrix. The feature representation capacity of a single RBM is limited, while this capacity is greatly enhanced when more RBMs are stacked, forming a DBN.

Figure 3.1 exhibits a schematic comparison of an autoencoder model and an RBM model.

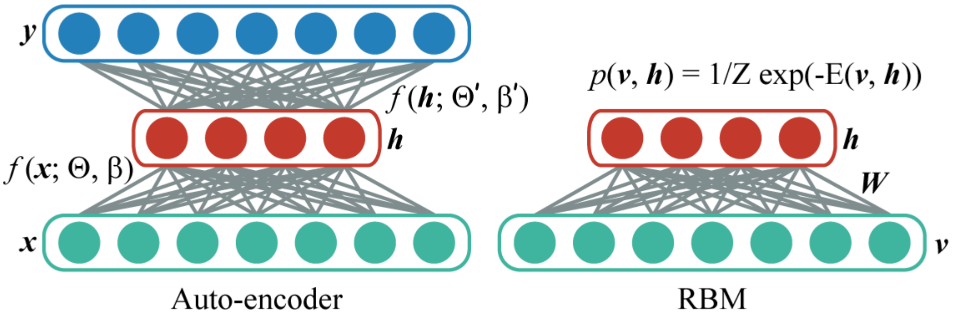


Figure 3.1: A schematic comparison of an autoencoder (left) and an RBM (right).

- **Applications in Remote Sensing**

In the field of hyper/multi-spectral image classification, the authors of [1] train a stacked autoencoder to extract hierarchical features from images for the purpose of classification. Subsequently, a DBN is evaluated in [2] in terms of unsupervised feature learning for hyperspectral image classification. In [3], the authors make use of a sparse autoencoder model to achieve unsupervised feature learning, and the learned features are subsequently fed into a support vector machine (SVM) to classify hyperspectral images.

Besides classification tasks, [4] designs a cascade antoencoder consisting a marginalized denoising autoencoder and a non-negative sparse autoencoder to address the problem of hyperspectral image unmixing. For the same task, the authors of [5] first learn a conventional stacked autoencoder on hyperspectral data as a pretrained model and then employ a variational autoencoder to jointly estimate endmembers and abundance fractions.

In addition, some detection tasks in hyper/multi-spectral images such as change detection [6–14] and anomaly detection [15] also benefit from the unsupervised feature learning mechanism of these models.

3.1.2 Convolutional Neural Networks (CNNs)

- **Models**

Since hyper/multi-spectral image analysis mainly involves CNNs for image classification in computer vision, several important CNN architectures are introduced first.

AlexNet [16]: In 2012, AlexNet significantly outperformed all other competitors and won the ImageNet challenge by achieving a top-5 error of 15.3%. In contrast, the second place, which is not a CNN-based method, only got a top-5 error of 26.2%. The general architecture of AlexNet, as shown in Figure 3.2 is similar to the classic LeNet-5 but considerably larger. The success of AlexNet has got much attention in the computer vision community and inspired researchers to take a serious look at deep learning for vision tasks. The main marks of AlexNet are as follows:

1. It is composed of 5 convolutional layers and 3 fully connected layers.
2. It introduces rectified linear unit (ReLU) as activation function, which now becomes the most widely used activation function in CNNs.
3. It is trained simultaneously on two Nvidia Geforce GTX 580 GPUs.
4. It implements Dropout layers to avoid the problem of overfitting.
5. It employs data augmentation techniques to greatly increase the number of training samples.

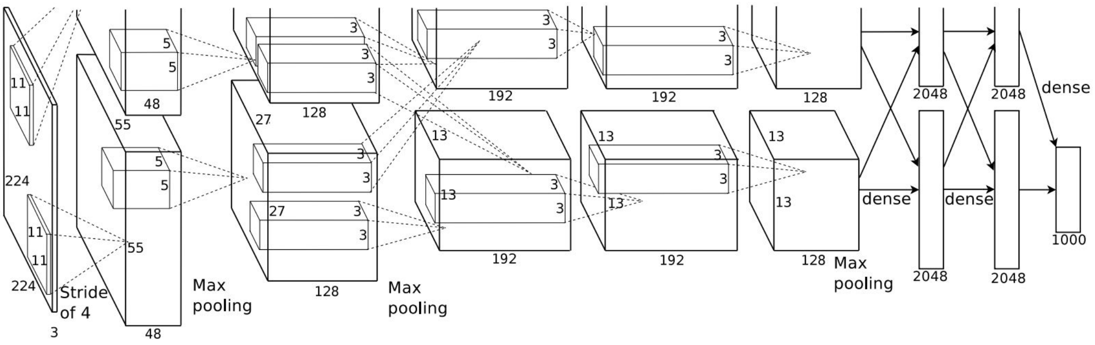


Figure 3.2: The architecture of AlexNet [16].

VGGNet [17]: The VGG networks, including VGG-16 and VGG-19, were introduced in 2014 and exhibit a deeper yet simpler CNN architecture. It was the runner-up at the ImageNet challenge in 2014 but now becomes one of the most preferred choices in the community for extracting features from images. It has addressed one important aspect of CNN architecture design – depth. Below are main points of VGGNet.

1. It uses convolutional filters with a very small receptive field of 3 × 3.
2. The spatial resolution is preserved in the same convolution block.

GoogLeNet (Inception) [18]: In 2014, GoogLeNet (a.k.a. Inception v1) was proposed and won the ImageNet Challenge with a top-5 error of 6.67%. It is independently developed in parallel to VGGNet. GoogLeNet introduces the following significant ideas.

1. It proposes a novel block, termed as inception module, as a basic unit to comprise a network. In an inception module, a series of convolutions at different scales are performed and subsequently aggregated.
2. The insight of GoogLeNet is that a large, powerful network can be done by increasing network width (number of units at each level) instead of only depth.

The architecture of GoogLeNet is illustrated in Figure 3.3.

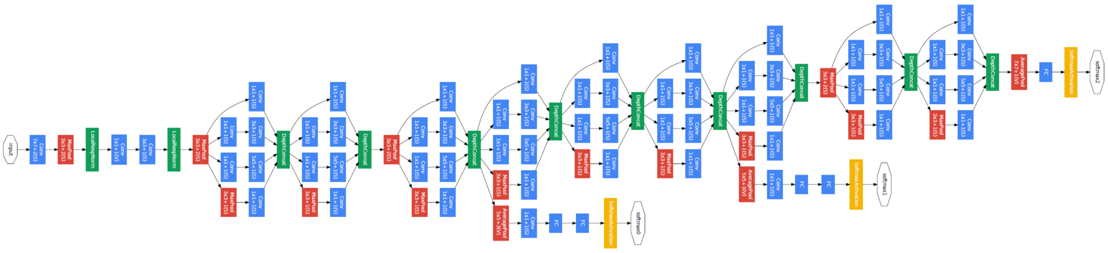


Figure 3.3: The architecture of GoogLeNet [18].

ResNet [19]: ResNet won the ImageNet challenge in 2015 with a top-5 error of 3.6% and set new records in both classification, detection, and localization through a single network architecture. Also the paper of ResNet won the best paper award of CVPR 2016. At that time, researches thought that building a larger CNN by increasing the number of layers would improve the accuracy of networks. But there are two problems, namely the vanishing gradient problem and the degradation problem. The former can be solved by regularization techniques like batch normalization, but the latter is hard to handle. ResNet was proposed to address the degradation problem (deeper networks lead to higher training errors). Figure 3.4 illustrates the architecture of ResNet, and the main contributions of ResNet are as follows:

1. It proves that optimizing a residual mapping $H(x) = F(x) - x$ is easier than optimizing an original, unreferenced mapping $F(x)$.

2. It creates a residual block, which is a basic unit of ResNet, by adding shortcut connections in an original convolutional block.

3. It enables the development of much deeper networks (hundreds of layers as opposed to tens of layers), e.g., ResNet-152.

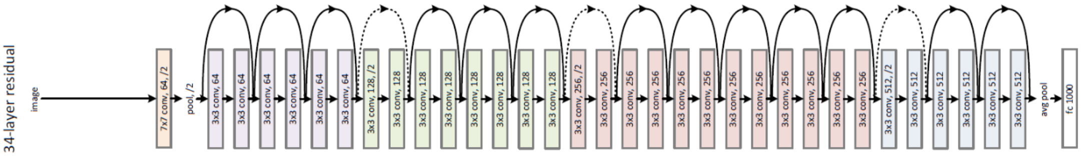


Figure 3.4: The architecture of ResNet [19].

DenseNet [20]: The idea behind DenseNet is: it may be useful to reference feature maps from earlier in the network. The paper of DenseNet received the best paper award of CVPR 2017. Overall, the architecture of DenseNet (see Figure 3.5) looks similar to that of ResNet, but has two important differences:

1. It concatenates feature maps instead of adding up them as ResNet does.

2. It adds skip connection from every previous layer.

3. Its layers are quite narrow, e.g., 12 filters per layer. In this way, it uses 3x less parameters as compared to ResNet for the similar number of layers.

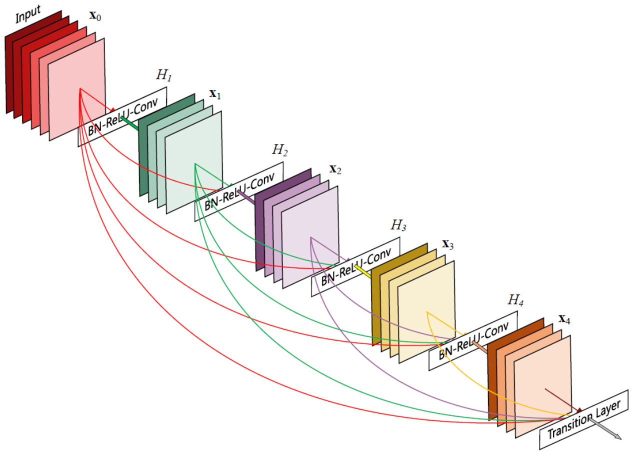


Figure 3.5: The architecture of DenseNet [20].

SENet [21]: An ensemble of SENets (Squeeze-and-Excitation Networks) with standard multi-scale and multi-crop fusion strategies obtained a top-5 error of 2.3% at ImageNet challenge in 2017. SENet cares about channel dependencies. The main points of SENet are as follows:

1. It proves that not all feature maps contribute equally, and the representation capacity of a network can be improved by emphasizing useful channels and suppressing less informative ones.

2. To this end, it designs a SE-block, which can be used in a plug-and-play fashion with any standard architectures.

C3D [22]: C3D was proposed for video classification tasks and uses 3D convolutions on video volumes. The contributions of C3D are:

1. It repurposes 3D CNNs as feature extractors.

2. It extensively searches for the best 3D convolutional kernel and architecture.

3. It makes use of deconvolutional layers to interpret model decisions.

The following figure shows the architecture of C3D network.

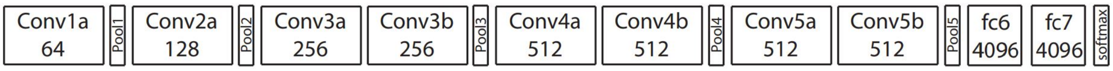


Figure 3.6: The architecture of C3D network [22].

- **Applications in Remote Sensing**

In the field of hyper/multi-spectral image analysis, CNNs are first used for classification tasks. The first try can be found in [23], where the authors train a very simple 1D CNN, which has only one convolutional layer, for hyperspectral image classification. Later, [24] employs a 2D CNN to extract both spatial and spectral information, and then the learned spectral-spatial features are fed into a multilayer perceptron (MLP) to perform the actual classification. The authors of [25] compare 1D CNN and 2D CNN in a practical application, crop type classification, and summarize that in terms of quantitative accuracy, 2D CNN is better, but the former easily leads to oversmoothed classification maps where small objects are usually misclassified.

2D CNNs have now become the mainstream for hyper/multi-spectral data analysis, as they can take full advantage of spectral-spatial information, and over the last years, the development of 2D CNN architectures in computer vision provides insights for this direction. For example, in [26], the authors are inspired by the idea of ResNet and present a spectral residual block and a spatial residual block in order to learn useful features from both spectral domain and spatial context. The authors of [27] introduce a residual learning-based Conv-Deconv network for the purpose of unsupervised spectral-spatial feature learning. [28] studies feature fusion in a ResNet for hyperspectral classification. More specifically, the authors first build a ResNet with three residual blocks and then fuse outputs of these three blocks, in order to obtain a better feature representation. Following the idea of DenseNet, the authors of [29] evaluate the performance of a DenseNet in hyperspectral image classification tasks. [30] constructs a densely connected CNN with 3D convolutional layers instead of 2D ones to extract spectral-spatial features. In [31], the authors present a modified DenseNet for hyperspectral data classification tasks, in which 3D atrous convolutional layers are used to learn features at different scales, and then the learned feature maps are densely connected in a DenseNet framework.

Hyper/multi-spectral images actually have a 3D data structure. Hence 3D CNNs, which have been widely used in video analysis tasks [32–44], give researchers an incentive. For 3D CNNs in video tasks, the third dimension is the time axis, while it refers to spectral bands in hyper/multi-spectral data analysis. The essential difference between 2D and 3D convolutions is that 2D convolutions use the same weights for the whole depth of the stack of spectral maps (multiple channels) and result in a single feature map, whereas 3D convolutions use 3D filters and produce a 3D volume as a result of the convolution, thus preserving spectral information. In [45], the authors evaluate a 3D CNN on the task of hyperspectral image classification. As compared to other networks, the used 3D CNN requires fewer parameters to achieve similar performance. The authors of [46] present a 3D CNN with a simple regularization.

To avoid overfitting, [47] jointly uses a dimension reduction method and a 2D CNN for spectral-spatial feature extraction. In [48], the authors first exploit a computational intelligence (particle swarm optimization) method to choose informative spectral bands and then train a 2D CNN using the selected bands. In [49], to properly train a CNN with limited ground truth data, the authors devise a CNN that takes as input a pair of hyperspectral pixels. By doing so, the amount of training data is greatly augmented.

Regarding unsupervised feature learning via CNNs, [50] presents a CNN to address the problem of unsupervised spectral-spatial feature extraction and estimate network weights via a sparse learning approach in a greedy layer-wise fashion. [51] proposes a residual learning-based fully conv-deconv network, aiming at unsupervised spectral-spatial feature learning in an end-to-end manner.

Better classification network architectures from computer vision (e.g., ResNet [19] and DenseNet [20]) also provide new insights for hyperspectral image classification [51–53]. Moreover, the integration of networks and other traditional machine learning models, e.g., conditional random field (CRF) and active learning, has also got attention recently [54, 55].

3.1.3 Recurrent Neural Networks (RNNs)

- **Models**

Recurrent neural networks (RNNs) extend feedforward networks with loops in connections, being able to process sequential data. So far there have been three kinds of widely used RNN architectures.

Fully connected RNN: This is the traditional RNN model, and its equations are as follows:

$$\boldsymbol{h}_t = \begin{cases} 0 & \text{if } t = 0 \\ \varphi(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) & \text{otherwise} \end{cases}, \tag{3.2}$$

where $\boldsymbol{x}_t$ is the input at the t-th time step of a sequential data $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T)$. $\boldsymbol{h}_{t-1}$ denotes the hidden state computed at time $t-1$ and is fed back into the network at the next time step to calculate $\boldsymbol{h}_t$. φ is a nonlinear activation function, e.g., a hyperbolic tangent function or sigmoid function. In a fully connected RNN, the following equation is usually used to update the recurrent hidden state as shown in Eq. (3.2).

$$\boldsymbol{h}_t = \varphi(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1}), \tag{3.3}$$

where $\boldsymbol{W}$ and $\boldsymbol{U}$ are learnable weight matrices.

The fully connected RNN model suffers from short-term memory. If an input sequential data is long enough, it cannot efficiently carry information from earlier time steps to later ones and hence may leave out important information from the beginning. This is mainly because during the backpropagation procedure of the network, the vanishing gradient problem emerges.

LSTM (Long short-term memory) [56]: To address the problem of the traditional RNN model, LSTM was proposed. LSTM is a special type of recurrent hidden unit, which is able to learn long-term dependencies. As compared to a fully connected RNN that overwrites its memory at each time step in a fairly uncontrolled way, an LSTM-based RNN transforms its memory in the following way: by using specific learning mechanisms for which pieces of information to remember, which to update, and which to pay attention to. This helps it keep track of information over longer periods of time. To describe the LSTM model mathematically, it first creates three gates, namely an output gate, an input gate, and a forget gate, as follows:

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{oi}\boldsymbol{x}_t + \boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{oc}\boldsymbol{c}_t)\,, \tag{3.4}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i i\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ic}\boldsymbol{c}_{t-1})\,, \tag{3.5}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f i\boldsymbol{x}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{fc}\boldsymbol{c}_{t-1})\,. \tag{3.6}$$

where the $\boldsymbol{W}$ matrices are trainable weights. $\boldsymbol{c}$ is termed as memory cell. At each time step, it can be updated by adding new contents via the input gate and discarding parts of the present memory contents via the forget gate:

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_{ci}\boldsymbol{x}_t + \boldsymbol{W}_{ch}\boldsymbol{h}_{t-1})\,. \tag{3.7}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i i\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ic}\boldsymbol{c}_{t-1})\,. \tag{3.8}$$

Finally, the activation of the network at the current time step can be computed as follows:

$$\boldsymbol{h}_t = \boldsymbol{o}_t \tanh(\boldsymbol{c}_t)\,. \tag{3.9}$$

The following figure shows the graphic model of LSTM.

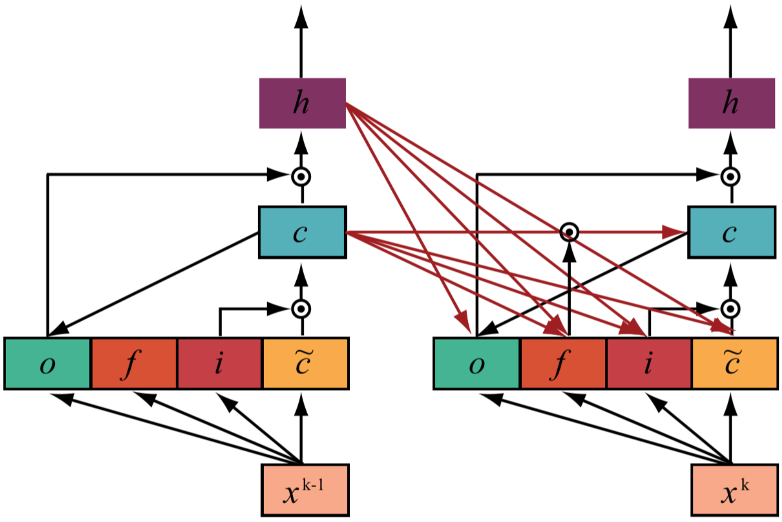


Figure 3.7: Graphic model of LSTM.

GRU (Gated recurrent unit) [57]: GRU is pretty similar to LSTM but has fewer parameters due to its simpler architecture. In comparison with LSTM, GRU gets rid of the memory cell and directly exposes the whole hidden state at each time. More specifically, it defines two gates, a reset gate and an update gate:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_{ri}\boldsymbol{x}_t + \boldsymbol{W}_{rh}\boldsymbol{h}_{t-1}) \,, \tag{3.10}$$

$$\boldsymbol{u}_t = \sigma(\boldsymbol{W}_{ui}\boldsymbol{x}_t + \boldsymbol{W}_{uh}\boldsymbol{h}_{t-1}) \,, \tag{3.11}$$

where the four $\boldsymbol{W}$ matrices are weight matrices. Then the activation at time step t can be calculated by a linear interpolation between the previous activation and a candidate activation: $\tilde{\boldsymbol{h}}_t$:

$$\boldsymbol{h}_t = (\boldsymbol{1} - \boldsymbol{u}_t)\boldsymbol{h}_{t-1} + \boldsymbol{u}_t\tilde{\boldsymbol{h}}_t \,, \tag{3.12}$$

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{W}\boldsymbol{x}_t) \,. \tag{3.13}$$

Figure 3.8 shows the architecture of PSPNet.

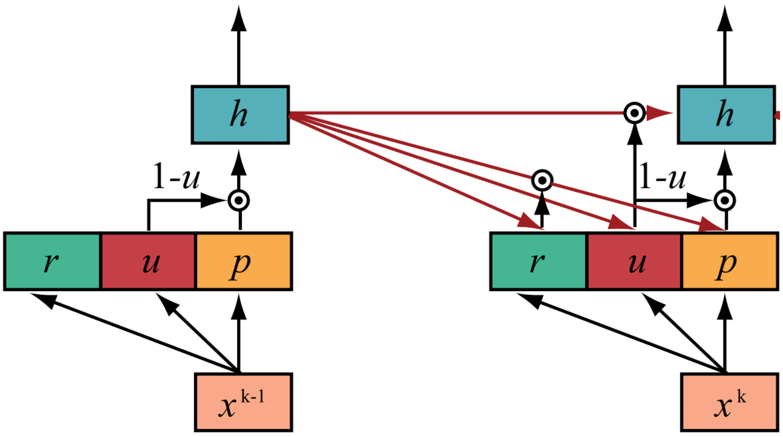


Figure 3.8: The architecture of GRU.

- **Applications in Remote Sensing**

Since RNNs are natural candidates for processing sequences [58–71], in remote sensing, they are mainly used in analyzing sequential remote sensing data. The first attempt in this direction can be found in [72], where the authors employ an LSTM-based RNN for detecting changes in bi-temporal multispectral images. Moreover, a transfer experiment is carried out to study the generalization capacity of a trained RNN-based change detection model. Later, by taking the spectrum as an ordered sequence, [73] makes use of a RNN to model the band-to-band variability and spectral correlation of hyperspectral data for the sake of classification. The authors of [74] train an LSTM-based RNN to classify crops from multi-temporal multispectral images. In change detection or the classification of multi-temporal image sequence, RNNs easily lead to noisy classification maps due to their lack of spatial content. To address this issue, [75] propose a recurrent convolutional neural network, which is composed of a convolutional sub-network and a recurrent sub-network, to learn a spectral-spatial-temporal feature representation for change detection tasks. In [76], the authors construct a similar network using convolutional recurrent layers for the purpose of classifying crops with multi-temporal data. The authors of [77] employ a temporal memory mechanism in a CNN to build a recurrent neural network structure, in order to predict clouds day and night in multi-temporal multispectral images.

Note that recurrent relation does not have to be over time, it can be over space for example. [78] is an example, where the authors make use of a fully connected RNN to model relationships between a given pixel and its neighbors, i.e., spatial relations.

3.2 Deep Learning for High-Resolution Aerial Images

This subsection mainly focuses on deep learning for semantic segmentation and introduces several classic semantic segmentation network architectures and applications in remote sensing.

- **Models**

FCN (Fully convolutional network) [79]: FCN is the first to develop an end-to-end trainable network for semantic segmentation tasks. More specifically, it convolutionalizes fully connected layers in a classification CNN (e.g., VGG-16) with kernels fully covering the whole spatial domain. Since now there is no longer the fully connected layer, the network can take images of arbitrary sizes. Then feature maps from different layers are upsampled using bilinearly initialized deconvolutions. Figure 3.9 illustrates the architecture of FCN. Below are a couple of key features of FCN:

1. It shows an excellent example for knowledge transfer from modern classification CNNs to performing dense prediction tasks like semantic segmentation.

2. It merges features from different stages in a classification CNN, which vary in coarseness of semantic information.

3. The upsampling of low-resolution feature maps is done by deconvolution.

SegNet [80]: Instead of reusing features learned in a classification CNN (called encoder) as in FCN, SegNet makes use of a decoder to learn segmentation maps with a desired full-resolution from low-resolution, high-level feature maps. SegNet has the following main points:

1. It employs unpooling to upsample feature maps in the decoder, in order to use and keep high frequency details as far as possible.

2. It doesn't use original fully connected layers in the encoder.

U-Net [81]: U-Net was initially proposed for medical image segmentation, but it has been successfully applied to a wide range of vision applications. As to its architecture

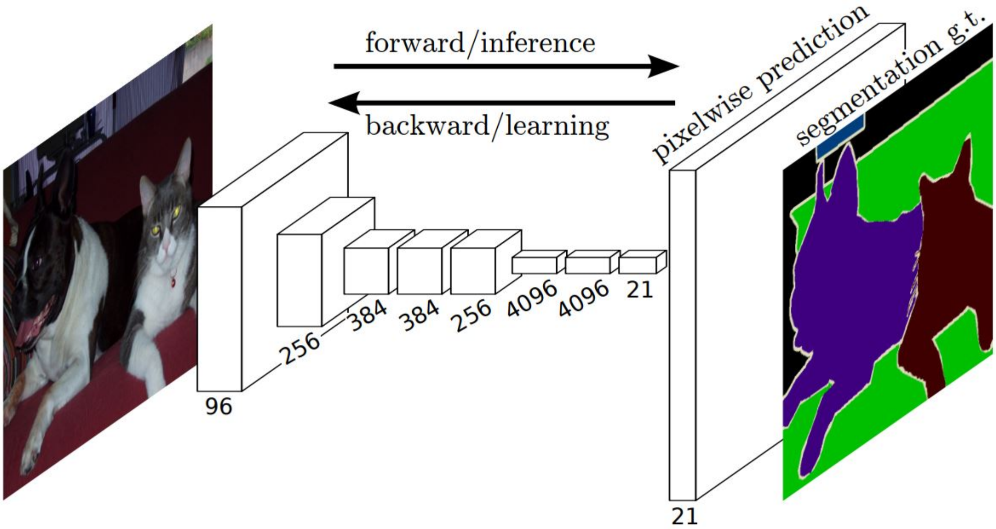


Figure 3.9: The architecture of FCN [79].

(see Figure 3.10), U-Net concatenates feature maps of the encoder to upsampled feature maps in the decoder at each stage to form a ladder-like structure.

DeepLabv1/v2/v3 [82]: Deeplabv1/v2 exploits dilated convolutions to enlarge the receptive field of the encoder. Moreover, the output is fed into a fully connected conditional random feild (CRF) model to produce final segmentation results. Deeplabv3 revisits the Deeplab framework and proposes to combine cascaded and parallel dilated convolutions modules.

PSPNet [83]: PSPNet (Pyramid Scene Parsing Network) also uses dilated convolutions by modifying the backbone ResNet architecture and has the following key features:

1. It introduces an auxillary loss at intermediate layers.
2. It devises a spatial pyramid pooling at top of the encoder to aggregate global context.

Figure 3.11 shows the architecture of PSPNet.

- **Applications**

Inspired by successes in the computer vision community [84–95], many researches are focusing on applying deep learning-based methods for semantic segmentation of aerial images. As an early trial [96], Sherrah employs a fully convolutional network (FCN) [79], which is pre-trained on natural images, and fine-tunes it on high resolution aerial images for predicting semantic labels of each pixel. Marmanis et al. [97] takes not only color images but also digital surface models (DSMs) data into account and employs a late fusion approach with two structurally identical, parallel FCNs, to fuse semantic information contained in both sources. Similarly, Audebert et al. [98] make use of SegNet [80, 99]

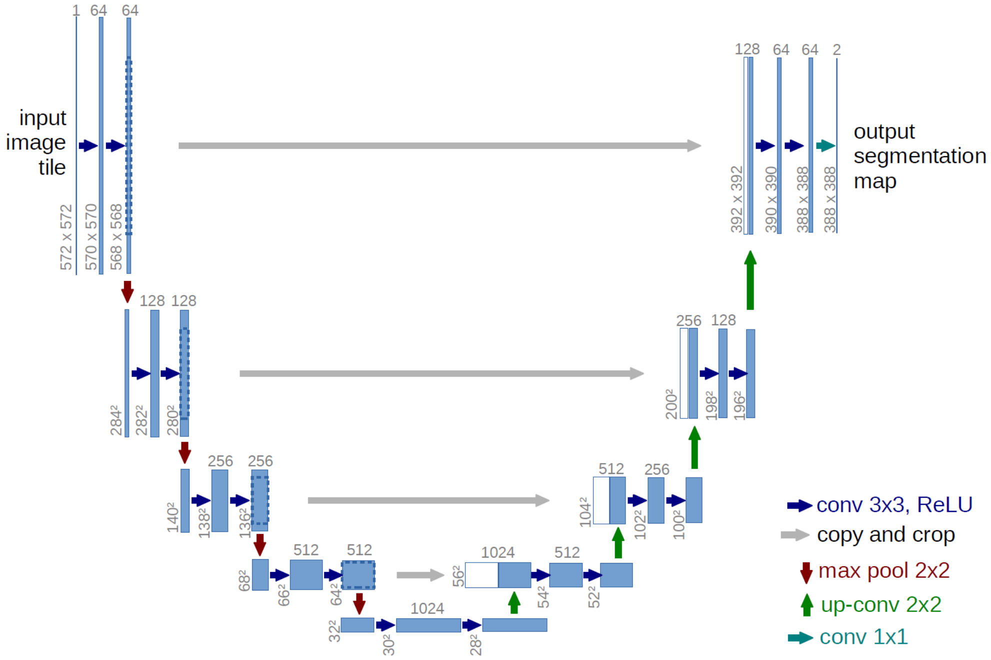


Figure 3.10: The architecture of U-Net [81].

with a residual correction to fuse optical images and DSM for semantic segmentation of high resolution aerial images. Later, in [100], Audebert et al. demonstrate that late fusion enables CNNs to recover errors stemming from ambiguous data while early fusion improves the efficiency of learning features jointly. However, fusing multimodal remote sensing data in an early fashion may suffer from higher sensitivity to missing data. In order to effectively fuse intermediate features, Maggiori et al. [101] introduce a multi-layer perceptron (MLP) on top of a base FCN and achieve satisfactory segmentation results. Moreover, in [102], Kellenberger et al. observe that spatial arrangements of many cities are similar, and such distributions can be learned from OSM data. Inspired by this observation, they cluster OSM building footprints over studied cities into different groups, which are then used to estimate prior distributions of each class. Afterwards, a conditional random field (CRF) is employed to combine learned spatial class prior and a CNN network for the final prediction.

In addition to focusing on multimodal data fusion, some studies deploy their efforts on exploiting semantic boundaries between different classes for semantic segmentation. Marmanis et al. [103] propose a two-step model, which consists of 1) learning a CNN for predicting edge likelihoods at multiple scales from color-infrared (CIR) and height data separately and 2) training another segmentation network with predicted boundaries as an extra input channel for the final semantic segmentation. The intuition behind this work is that using predicted boundaries helps to achieve sharper segmentation maps. In [104], Volpi and Tuia employ a multi-task CNN to predict not only semantic labels but

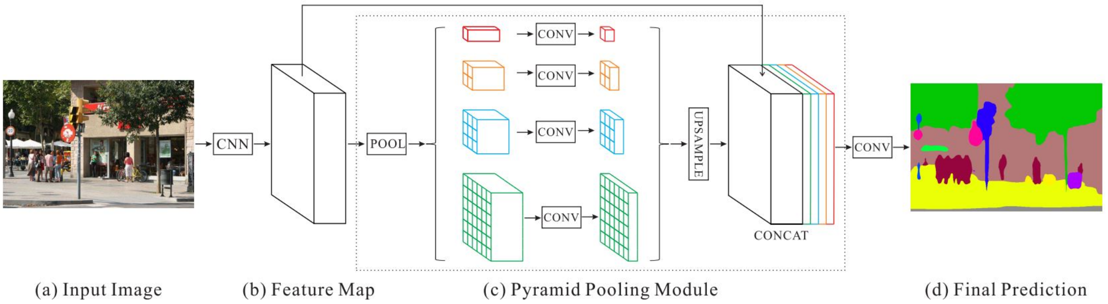


Figure 3.11: The architecture of PSPNet [83].

also class boundaries, which are then used to partition images into hierarchical regions. Among them, consistent regions are parsed with a CRF to reduce the complexity of output segmentation maps.

Besides, researches such as [105] and [106] pay attention to the precise identification of small objects. Specifically, Kampffmeyer et al. [105] train an FCN with the cross-entropy loss function weighted with median frequency balancing, which is proposed by Eigen and Fergus [107] for quantifying the uncertainty at a pixel level, and achieve good accuracy for all objects, especially small ones. Audebert et al. [106] propose a deep-learning-based "segment-before-detect" method for semantic segmentation and subsequent classification of several types of vehicles in high-resolution remote sensing images. The use of SegNet [80] in this method is capable of producing pixel-wise annotations for vehicle semantic mapping. In addition, several recent works in the semantic segmentation of high-resolution aerial imaging also involve vehicle segmentation.

Furthermore, several researches are conducted on studying the properties of segmentation networks. In [108], Volpi and Tuia perform a comparison between SegNet and a standard CNN, where patch classification is conducted. In [109] Marcos et al. propose a segmentation network architecture called rotation equivariant vector field network (RotEqNet) to encode rotation equivariance in the network itself. By doing so, the network can be confronted with a simpler task, than has to learn specific weights to address rotated versions of the same object class. In [110], Shivangi et al. assume that changes in height often rely on those in classes. Based on this assumption, they extend the semantic segmentation to a multi-task problem, and use a CNN to jointly conduct semantic segmentation and height estimation. Experimental results prove that the model performance of semantic segmentation has been increased compared to a single-task model. Considering that dense labels are time- and labor-consuming, Maggiolo et al. [111] propose a method, which consists of CRF, CNN, and clustering concepts, to improve the performance of CNN architectures when scribbled annotations are given. As demonstrated in the result, improvements can be seen on the overall accuracy and precision, while the recall is slightly reduced. To address this, they suggest that more advanced clustering methods can be considered in further researches. Wurm et al. [112] analyze transfer learning capabilities of FCNs for semantic segmentation of different satellite images. They trained a model on high-resolution satellite images from QuickBird and then transfer this model to low resolution images from Sentinel-2 and TerraSAR-X. Experiments demonstrate that the transfer learning can significantly improve the performance on Sentinel-2 data.

3.3 Attention Mechanism in Deep Learning

When training a network, sometimes we expect it to be able to focus on important parts of an image instead of the whole image. One way of accomplishing this is through attention mechanisms. In machine learning, "attention" refers to a group of techniques that help a "model-in-training" notice important things more effectively.

The gating mechanism, which is similar to the attention mechanism, has been widely used in modeling and processing temporal sequences. E.g., long short-term memory (LSTM)-based networks [56, 113] harness three gates to cope with vanishing gradients. Similarly, a gated recurrent unit (GRU) [114, 115] is designed to implement the modulation of information flow through the gating mechanism.

In natural language processing (NLP), [116] applies the attention mechanism to images to generate captions. A given image is first encoded by a CNN to extract features. Then an LSTM decoder consumes the convolution features to produce descriptive words one by one, where the weights are learned through attention. The visualization of the attention weights demonstrates which regions of the image the model is paying attention to so as to output a certain word. This work first proposes the distinction between "soft" vs. "hard" attention, based on whether the attention has access to the entire image or only a patch. In [117], the authors propose the "global" and "local" attention. The former is similar to the soft attention, while the latter is an interesting blend between hard and soft, an improvement over the hard attention to make it differentiable: the model first predicts a single aligned position for the current target word and a window centered around the source position is then used to compute a context vector.

In addition, several recent works in computer vision have shown the benefit of introducing the gating mechanism to vision problems. To name a few, [118] proposes a gating mechanism that is capable of dynamically balancing contributions of the current event and its surrounding contexts in their model for dense video captioning tasks. In [21], the authors build a gated block for image classification tasks and demonstrate its good performance on large-scale image recognition. [119] addresses person re-identification tasks through utilizing a network module based on a soft gating mechanism, which enables the network to concentrate on significant local regions of an input image pair adaptively.

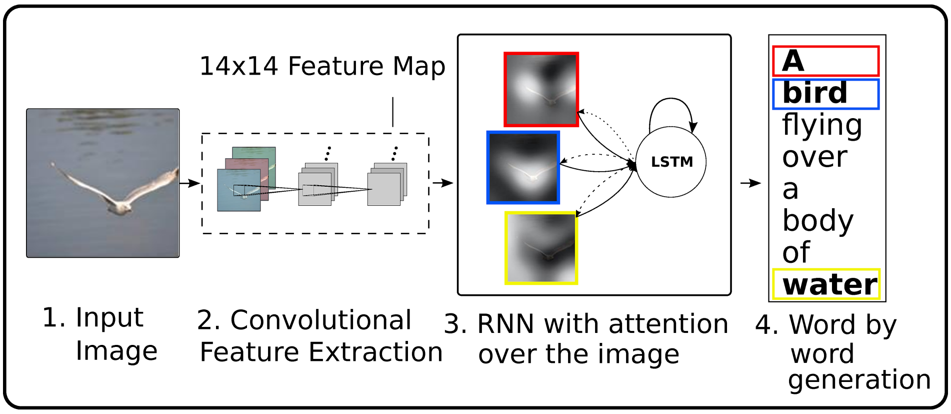

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 3.12: An attention-based network for image captioning [116].

3.4 Relational Reasoning Networks

A relation reasoning network is an artificial neural network component with a structure that can reason about relations among objects. An example category of such relations is spatial relations (above, below, left, right, in front of, behind).

Recently, the authors of [120] propose a relational reasoning network for the problem of visual question answering (an example in remote sensing can be found in [121]), and this network achieves a super-human performance. The architecture of this relational reasoning network can be seen in Figure 3.13. Later, [122] proposes a temporal relation network to enable multi-scale temporal relational reasoning in networks for video classification tasks. In [123], the authors propose an object relation module, which allows modeling relationships among sets of objects, for object detection tasks. Our work is motivated by the recent success of these works, but we focus on modeling spatial and channel relations in a CNN for semantic segmentation.

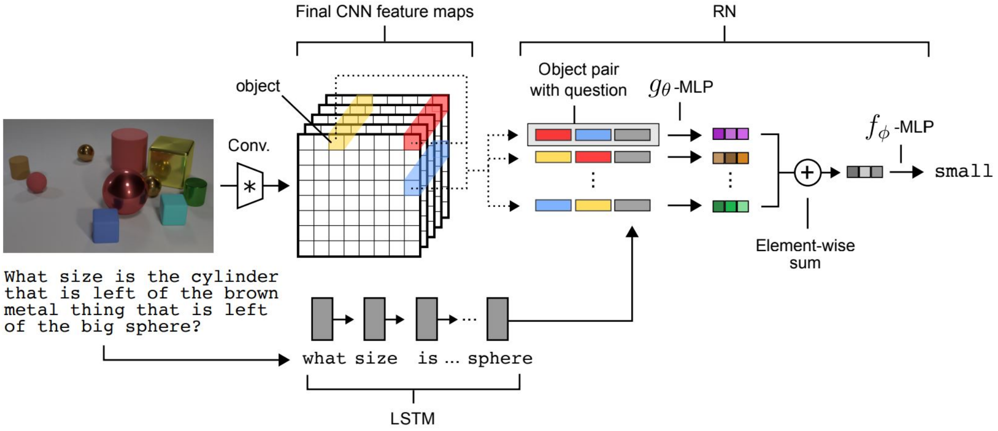

Final CNN feature maps
RN
object
Conv.
Object pair with question
g_θ-MLP
f_ϕ-MLP
small
Element-wise sum
What size is the cylinder that is left of the brown metal thing that is left of the big sphere?
what size is ... sphere
LSTM

Figure 3.13: The architecture of relational reasoning network in [120].

4 Summary of the Work

4.1 Perception: Modeling Hyperspectral Data from a Sequential Perspective

4.1.1 Motivation

Hyperspectral data can be seen as a set of orderly and continuing spectra sequences in the spectral space. Analyzing hyperspectral imagery from a sequential perspective has not been addressed so far. Our motivation in this work is to explore the representation of hyperspectral pixels via the sequential perspective. The RNN exploits a recurrent procedure to characterize spectral correlation and band-to-band variability, where the network parameters are determined by training with available samples. In this context, we propose a novel RNN with a specially designed activation function and modified GRU to solve the multiclass classification for hyperspectral imagery.

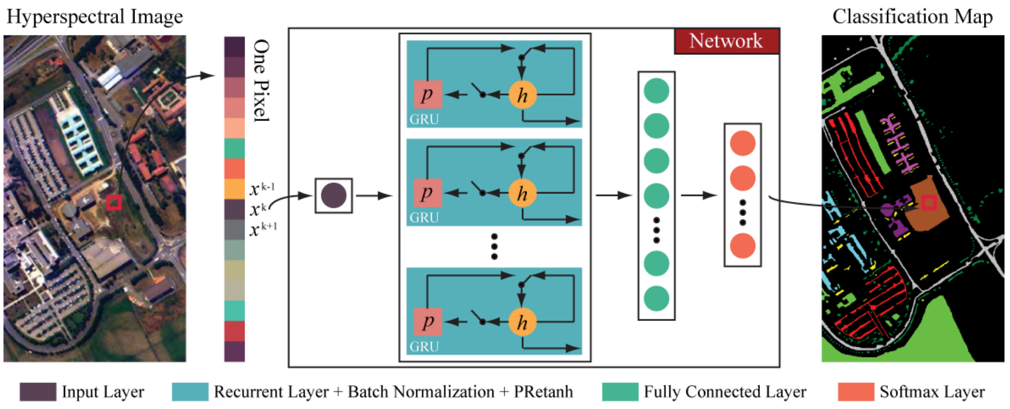


Figure 4.1: Overview of the RNN for hyperspectral image classification.

4.1.2 Methodology

In the main procedure of the proposed recurrent network, as illustrated in Figure 4.1, the input of the network is a hyperspectral pixel $\mathbf{x}$, where the k-th spectral band is denoted as x^k. The output is a label that indicates the category the pixel belongs to.

The entire classification map can be obtained by applying the network to all pixels in the image. The flowchart of our RNN can be summarized as follows:

1. First, the value of the existing spectral band x^k is fed into the input layer.
2. Then, the recurrent layer receives x^k and calculates the hidden state information for the current band; it also restores that information in the meantime.
3. Subsequently, the value of the next band x^{k+1} is input to the recurrent layer simultaneously with the state information of x^k, and the activation at spectral band $k+1$ is computed by a linear interpolation between proposal activation and the activation of the previous band k.
4. Finally, the RNN predicts the label of the input hyperspectral pixel by looping through the entire hyperspectral pixel sequence.

Two important factors affect the performance of RNN: the activation function and the structure of the recurrent unit. In the next section, we will discuss our innovative contributions on these two factors in detail.

Parametric Rectified tanh (PRetanh). We introduce a newly defined activation function – parametric rectified tanh (PRetanh). It is defined as

$$f(h_i) = \begin{cases} \tanh(h_i) & \text{if } h_i > 0 \\ \lambda_i \tanh(h_i) & \text{if } h_i \leq 0 \end{cases}, \tag{4.1}$$

where h_i is the input of the nonlinear activation f on the i-th channel and $0 \leq \lambda_i \leq 1$ is a coefficient that can control the range of the negative part. The subscript i means that PRetanh can be varied in different channels. When $\lambda_i = 0$, it turns to

$$f(h_i) = \max(0, f(h_i)) = \max(0, \tanh(h_i)) \,. \tag{4.2}$$

When λ_i is a learnable parameter, we refer to Eq. (4.1) as a parametric rectified hyperbolic tangent function. Eq. (4.1) is equivalent to

$$f(h_i) = \max(0, \tanh(h_i)) + \lambda_i \min(0, \tanh(h_i)) \,. \tag{4.3}$$

In our method, the PRetanh parameter λ_i is adaptively learned jointly with the whole neural network model. We expect that end-to-end training can lead to more specialized activations. Note that one extra parameter is introduced in PRetanh. The total number of extra parameters for each layer is equal to the number of channels, which is negligible when taking into account the number of weights of the whole network. Therefore, we anticipate no extra risk of overfitting with the same number of training samples. In addition, a channel-shared variant version of PRetanh can be considered:

$$f(h_i) = \max(0, \tanh(h_i)) + \lambda \min(0, \tanh(h_i)) \,, \tag{4.4}$$

where all channels of one layer share the same coefficient λ. In this case, only a single extra parameter is introduced for each layer.

Gated Recurrent Unit with PReLU. A gated recurrent unit can cause a recurrent unit to adaptively capture the dependencies of different spectral bands. Similar to the LSTM unit, the gated recurrent unit has gate units that control the flow of information inside the unit without including separate memory cells.

The activation h_i^k of the i-th gated recurrent unit at spectral band k is computed by a linear interpolation between the proposal activation p_i^k and the activation of the previous spectral band h_i^{k-1}:

$$h_i^k = u_i^k p_i^k + (1 - u_i^k) h_i^{k-1} \,, \tag{4.5}$$

where u_i^k is an update gate that determines how much the unit updates its activation or content. The update gate u_i^k can be calculated as follows:

$$u_i^k = \sigma(\mathbf{w}_{ui} x^k + \mathbf{W}_{uh} \mathbf{h}^{k-1})_i \,, \tag{4.6}$$

where $\mathbf{w}_{ui}$ is the input-update weight vector and $\mathbf{W}_{uh}$ represents the update-hidden weight matrix.

Similarly to LSTM, the gated recurrent unit takes a linear sum between the newly computed state and the present state. However, it lacks a mechanism to control what part of the state information will be exposed, rather exposing the whole state value at each spectral band.

The proposal activation p_i^k is computed using the value of the existing spectral band and the activation of the previous band, which reflects the updated information of the recurrent hidden state. It is calculated with PRetanh and batch normalization as follows:

$$p_i^k = f(g(\mathbf{w}_{pi} x^k + \mathbf{W}_{rh}(\mathbf{r}^k \odot \mathbf{h}^{k-1})))_i \,, \tag{4.7}$$

where $\mathbf{r}^k$ is a set of reset gates, $\mathbf{w}_{pi}$ denotes the proposal-input weight vector, and $\mathbf{W}_{rh}$ represents the reset-hidden weight matrix. Moreover, $f(\cdot)$ and $g(\cdot)$ represent PRetanh and batch normalization, respectively. When the reset gate r_i^k is fully off, i.e., r_i^k is 0, it will completely discard the activation of the hidden layer at previous spectral bands h_i^{k-1} and only use the value of the existing spectral band x^k. When open (r_i^k close to 1), in contrast, the reset gate will partially keep the information of the previous step.

Let $\hat{p}_i^k = \mathbf{w}_{pi} x^k + \mathbf{W}_{rh}(\mathbf{r}^k \odot \mathbf{h}^{k-1})$. Eq. (4.7) can then be transformed as

$$\begin{aligned} p_i^k = & \max(0, \tanh(\alpha_i \frac{\hat{p}_i^k - \mathrm{E}[\hat{p}_i^k]}{\sqrt{\mathrm{Var}[\hat{p}_i^k]}} + \beta_i)) \\ & + \lambda_i \min(0, \tanh(\alpha_i \frac{\hat{p}_i^k - \mathrm{E}[\hat{p}_i^k]}{\sqrt{\mathrm{Var}[\hat{p}_i^k]}} + \beta_i)) \,, \end{aligned} \tag{4.8}$$

The reset gate r_i^k is computed similarly to the update gate:

$$r_i^k = \sigma(\mathbf{w}_{ri} x^k + \mathbf{W}_{rh} \mathbf{h}^{k-1})_i \,, \tag{4.9}$$

where $\mathbf{w}_{ri}$ and $\mathbf{W}_{rh}$ are the reset-input weight vector and the reset-hidden weight matrix, respectively.

Figure 4.2 shows the graphic model of the gated recurrent unit though time.

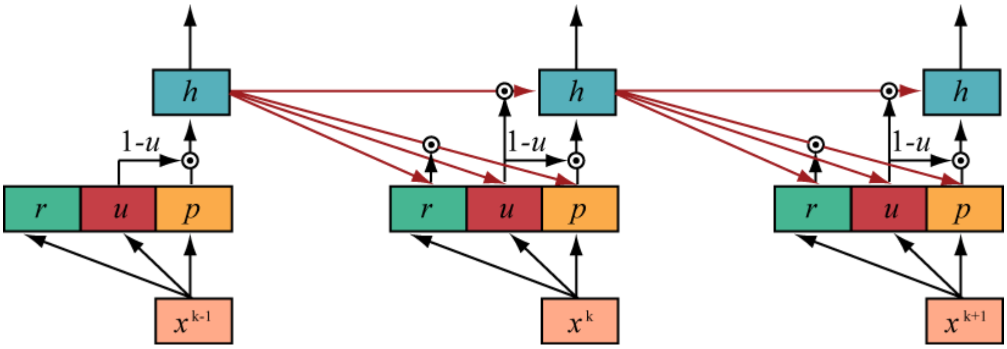


Figure 4.2: Graphic model of a GRU through time.

4.1.3 Results

Dataset. Pavia University dataset is acquired by reflective optics system imaging spectrometer (ROSIS). The image is of 610 × 340 pixels covering the Engineering School at the University of Pavia, which was collected under the HySens project managed by the German Aerospace Agency (DLR). The ROSIS-03 sensor comprises 115 spectral channels ranging from 430 to 860 nm. In this data set, 12 noisy channels have been removed and the remaining 103 spectral channels are investigated in this paper. The spatial resolution is 1.3 m per pixel. The available training samples of this data set cover nine classes of interests.

Quantitative Evaluation. Table 4.1 shows the quantitative comparison with other methods.

For more experimental results and technical details, please refer to Appendix A.

Table 4.1: The Classification Accuracies of Different Techniques in Percentages for Pavia University. The Best Accuracy in Each Row Is Shown in Bold.

Class Name	RF	SVM	CNN	LSTM	GRU	**GRU-PRetanh**
Asphalt	80.85	80.80	83.73	77.45	78.42	**84.45**
Meadows	55.29	66.78	65.70	61.83	69.17	**85.24**
Gravel	52.93	**73.18**	67.03	64.60	47.83	54.31
Trees	**98.79**	95.17	94.03	97.98	97.16	95.17
Metal Sheets	99.26	99.55	99.41	99.18	97.84	**99.93**
Bare Soil	78.76	92.90	**96.30**	91.19	85.86	80.99
Bitumen	84.36	90.08	**93.83**	90.90	86.84	88.35
Bricks	91.58	91.20	93.56	92.29	**94.27**	88.62
Shadows	98.20	93.77	99.79	97.47	94.93	**99.89**
OA	71.37	78.82	79.27	75.92	77.70	**84.99**
AA	82.23	87.05	**88.15**	85.88	83.59	86.33
Kappa	0.6484	0.7358	0.7423	0.7028	0.7201	**0.8048**

4.2 Perception: Unsupervised Feature Learning with Abundant Bands

4.2.1 Motivation

Despite the big success of the supervised CNNs or RNNs, they have at least one potential drawback detailed as follows: there is a need for a good supply of labeled training samples to be used for supervised training. However, these are difficult to collect, and there are diminishing returns of making the labeled data set larger and larger. In other words, the supervised CNNs generally suffer from either small number of training samples or imbalanced data sets.

Hence, unsupervised spectral-spatial feature learning, which has a quick access to arbitrary amounts of unlabeled data, is conceptually of high interest. In general, the main purpose of unsupervised feature learning is to extract useful features from unlabeled data, to detect and remove input redundancies, and to preserve only essential aspects of the data in robust and discriminative representations. In a pioneer work moving from the supervised CNN to unsupervised CNN, Romero et al. [124] proposed an unsupervised convolutional network for learning spectral-spatial features using sparse learning to estimate the weights of the network. However, this model was trained in a greedy layer-wise fashion, i.e., it is not an end-to-end network. In this chapter, we aim to propose an end-to-end network for unsupervised spectral-spatial feature learning of hyperspectral imagery.

4.2.2 Methodology

Denote by $(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y})$ random variables representing a 3D hyperspectral patch, its encoded feature representation, and the reconstructed output. The joint probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$ can be described as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})\,, \tag{4.10}$$

where $p(\boldsymbol{x})$ is the distribution of 3D hyperspectral patches and $p(\boldsymbol{y}|\boldsymbol{x})$ is the distribution of reconstructed outputs given the hyperspectral patches. Thus the conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ can be written as

$$p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y}, \boldsymbol{h}|\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{h})p(\boldsymbol{h}|\boldsymbol{x})\,, \tag{4.11}$$

where $p(\boldsymbol{h}|\boldsymbol{x})$ indicates the distribution of the encoded feature representations given the input hyperspectral patches. As a special case, $\boldsymbol{y}$ may be a deterministic function of $\boldsymbol{x}$. Ideally we would like to find $p(\boldsymbol{h}|\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{h})$, but direct application of Bayesian theory is not feasible. We, therefore, in this work resort to an estimate function $f(\boldsymbol{x})$ which minimizes the following mean squared error objective:

$$\mathbb{E}_{\boldsymbol{x}}\|\boldsymbol{x} - f(\boldsymbol{x})\|_2^2\,. \tag{4.12}$$

The minimizer of this loss is the conditional expectation:

$$\hat{f}(\boldsymbol{x}_0) = \mathbb{E}_{\boldsymbol{y}}[\boldsymbol{y}|\boldsymbol{h}] + \mathbb{E}_{\boldsymbol{h}}[\boldsymbol{h}|\boldsymbol{x} = \boldsymbol{x}_0]\,, \tag{4.13}$$

that is the expected reconstructed output given a hyperspectral patch.

Given a set of unlabeled 3D hyperspectral patches $\{\boldsymbol{x}_i\}$, we learn the weights $\boldsymbol{\Theta}$ of a network $f(\boldsymbol{x}; \boldsymbol{\Theta})$ to minimize a Monte-Carlo estimate of the loss (4.12):

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_i \|\boldsymbol{x}_i - f(\boldsymbol{x}_i; \boldsymbol{\Theta})\|_2^2\,. \tag{4.14}$$

This means that we train the network to reproduce the input results in learning high-level abstract features in an unsupervised manner.

Here, we propose a fully Conv-Deconv network (cf. Figure 4.3) in which the desired output is the input data itself. The proposed network architecture is composed of two parts, i.e., the convolutional sub-network and deconvolutional sub-network. The former corresponds to an encoder that transforms the input 3D hyperspectral patch $\boldsymbol{x}_i$ to abstract feature representation $\boldsymbol{h}_i$, whereas the latter plays the role of a decoder that reproduces the initial input data from the encoded feature. Each layer in the convolutional sub-network has a corresponding decoder layer in the deconvolutional sub-network.

But a problem arises when we attempt to directly train such a network. Although the network starts greatly reducing errors on both the training and validation samples

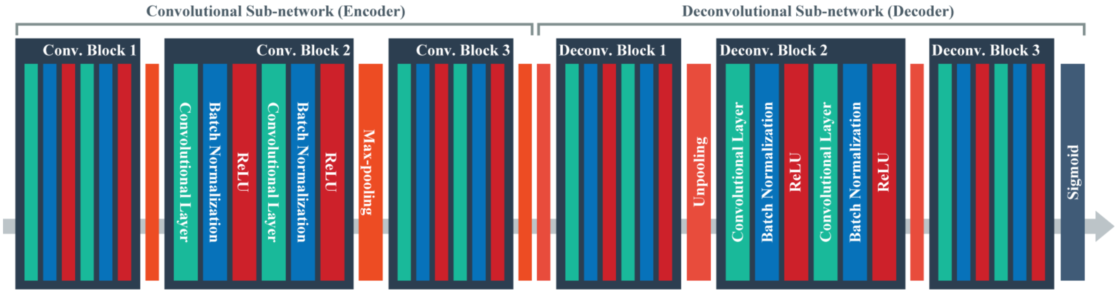


Figure 4.3: We propose a network architecture which learns to extract spectral-spatial features by reconstructing the initial input 3D hyperspectral patches, being trained end-to-end. There are no fully connected layers and hence it is fully Conv-Deconv network. The proposed network architecture is composed of two parts, i.e., convolutional sub-network and deconvolutional sub-network. The former corresponds to a encoder that transforms the input 3D hyperspectral patches to abstract feature representations, whereas the latter plays the role of decoder that reproduces the initial input data from the encoded features. Each layer in the convolutional sub-network has a corresponding decoder layer in the deconvolutional sub-network.

during the first few epochs, it rapidly converges to a fairly high value, which means the learning of the network is significantly slowed down and eventually gets stuck into a local minimum. This indicates that such network architecture is not easy to optimize. We think the obstacles to train the proposed fully Conv-Deconv network are as follows:

1. In the Conv-Deconv network, the exact copy of the input high dimensional 3D hyperspectral patch has to go through all layers until it reaches the output layer. With many weight layers, this becomes an end-to-end relation requiring very long-term memory. For this reason, the notorious vanishing gradient problem can be critical, which handicaps the learning process of the network.

2. The unpooling operation in the deconvolutional sub-network increases the spatial resolution of feature maps by simply adding zeros, which ignores the location of the maximum value in the receptive field of pooling layer, leading to loss of edge information during the decoding procedure. Without this detailed information, it is difficult for the optimizer to lead the network to better solutions.

To address the aforementioned problems, in this subsection, we refine the proposed fully Conv-Deconv network architecture by incorporating residual learning and a new unpooling operation that can use memorized max-pooling indices from the corresponding encoded feature maps and enables reconstruction to be more accurate. The new network architecture is shown in Figure 4.4.

The proposed Conv-Deconv network with residual learning is a modularized network architecture that stacks residual blocks. Similarly to the convolutional blocks, a residual block consists of several convolutional layers with the same feature map size and the same number of filters. However, it performs the following calculation:

$$\boldsymbol{\varphi}_l = g(\boldsymbol{\phi}_l) + \mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l)\,, \tag{4.15}$$

$$\boldsymbol{\phi}_{l+1} = f(\boldsymbol{\varphi}_l)\,. \tag{4.16}$$

Here, $\boldsymbol{\phi}_l$ indicates the feature maps that are fed into the l-th residual block and satisfies $\boldsymbol{\phi}_0 = \boldsymbol{x}$ where $\boldsymbol{x}$ is the input 3D hyperspectral patch. $\boldsymbol{\Theta}_l = \{\boldsymbol{\Theta}_{l,k} | 1 \leq k \leq K\}$ represents a collection of weights associated with the l-th residual block, and K denotes that there are K convolutional layers in a residual block. Moreover, $\mathcal{F}$ is the residual function and is generally achieved by few stacked convolutional layers. The function f indicates the activation function such as a linear activation function or ReLU, and f works after element-wise addition. The function g is fixed to an identity mapping: $g(\boldsymbol{\phi}_l) = \boldsymbol{\phi}_l$.

If f adopts a linear activation function and also acts as an identity mapping, i.e., $\boldsymbol{\phi}_{l+1} = \boldsymbol{\varphi}_l$, we can obtain the output of the l-th residual block by putting Eq. (4.15) into Eq. (4.16):

$$\boldsymbol{\phi}_{l+1} = \boldsymbol{\phi}_l + \mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l)\,. \tag{4.17}$$

In contrast, a convolutional block only performs the following computation:

$$\boldsymbol{\phi}_{l+1} = \mathcal{H}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l)\,. \tag{4.18}$$

Recursively like

$$\begin{aligned} \boldsymbol{\phi}_{l+2} =& \boldsymbol{\phi}_{l+1} + \mathcal{F}(\boldsymbol{\phi}_{l+1}; \boldsymbol{\Theta}_{l+1}) \\ =& \boldsymbol{\phi}_l + \mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l) + \mathcal{F}(\boldsymbol{\phi}_{l+1}; \boldsymbol{\Theta}_{l+1})\,, \end{aligned} \tag{4.19}$$

etc., we will get the following recurrence formula:

$$\boldsymbol{\phi}_L = \boldsymbol{\phi}_l + \sum_{i=l}^{L-1} \mathcal{F}(\boldsymbol{\phi}_i; \boldsymbol{\Theta}_i)\,, \tag{4.20}$$

for any shallower block l and any deeper block L.

As exhibited in Eq. (4.20), the network with residual learning has some interesting and nice properties: 1) The feature maps $\boldsymbol{\phi}_L$ of any deeper residual block L can be considered to be adding the feature maps $\boldsymbol{\phi}_l$ of any shallower block l and a residual function in a form of $\sum_{i=1}^{L-1} \mathcal{F}$, representing that the network is in a residual fashion and is capable of learning some new features between any blocks l and L; and 2) with both the g and f being identity mappings, i.e., $g(\boldsymbol{\phi}_l) = \boldsymbol{\phi}_l$ and $f(\boldsymbol{\varphi}_l) = \boldsymbol{\varphi}_l$, a network with residual learning creates a direct path for propagating information through the entire network, which can effectively avoid the vanishing gradient problem. These two respects are in contrast to the Conv-Deconv network equipped with common convolutional blocks in which the feature maps $\boldsymbol{\phi}_L$ are a set of matrix products, namely, $\prod_{i=0}^{L-1} \boldsymbol{\Theta}_i \boldsymbol{\phi}_0$.

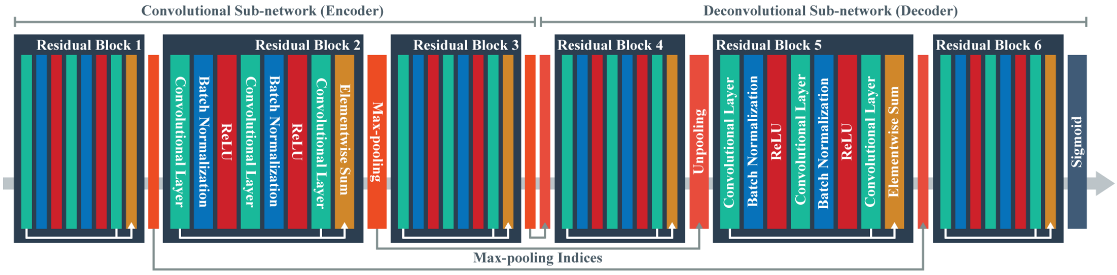


Figure 4.4: We refine the proposed fully Conv-Deconv network architecture by incorporating residual learning and a more appropriate unpooling operation, which can use memorized max-pooling indices from the corresponding encoded feature maps and enables reconstruction to be more accurate.

4.2.3 Results

Quantitative Evaluation. Table 4.2 shows the quantitative comparison with other methods on the Pavia University dataset.

Table 4.2: The Classification Accuracies of Different Techniques in Percentage for Pavia University. The Best Accuracy in Each Row Is Shown in Bold.

Class Name	RF	SVM	1D-CNN	2D-CNN	SICNN	Ours
Asphalt	80.94	84.84	83.73	69.25	**84.21**	78.99
Meadows	55.91	67.09	65.70	93.39	91.10	**97.16**
Gravel	53.26	**72.13**	67.03	63.13	64.36	61.46
Trees	**98.76**	95.72	94.03	94.39	95.53	95.76
Metal Sheets	99.11	99.48	99.41	**100**	97.70	97.77
Bare Soil	79.26	93.30	**96.30**	49.06	56.53	59.46
Bitumen	83.76	91.88	**93.83**	72.26	77.29	79.5
Bricks	91.06	92.56	93.56	94.32	95.57	**96.82**
Shadows	98.10	97.47	**99.79**	93.77	96.20	92.40
OA	71.66	79.88	79.28	82.66	85.25	**87.39**
AA	82.24	88.27	**88.15**	81.06	84.28	84.37
Kappa	0.6517	0.7487	0.7423	0.7688	0.8041	**0.8308**

For more experimental results and technical details, please refer to Appendix B.

4.3 Perception: Spectral-Spatial-Temporal Feature Learning for Multitemporal Analysis

4.3.1 Motivation

As an important branch of deep learning family, a recurrent neural network (RNN) is a natural candidate to tackle the temporal connection between multitemporal sequence data in change detection tasks. Recently, Lyu et al. [72] make use of an end-to-end RNN to solve the multi/hyper-spectral image change detection task, since RNN is well known to be good at processing sequential data. In their framework, a long short-term memory (LSTM)-based RNN is employed to learn a joint spectral-temporal feature representation from a bi-temporal image sequence. In addition, the authors also show the versatility of their network by applying it to detect multi-class changes and pointing out a good transferability for change detection in an “unseen” scene without fine-tuning. The authors of [74] follow a similar idea, where an RNN based on LSTM units is used to extract dynamic spectral-temporal features but, in contrast to the change detection scenario, their goal is to address land cover classification of multitemporal image sequences. However, we observe that RNNs always result in noisy scatter points in change detection maps. This is mainly because RNNs do not take spatial information into account. In this chapter, we learn joint spectral-spatial-temporal features using an end-to-end network for change detection.

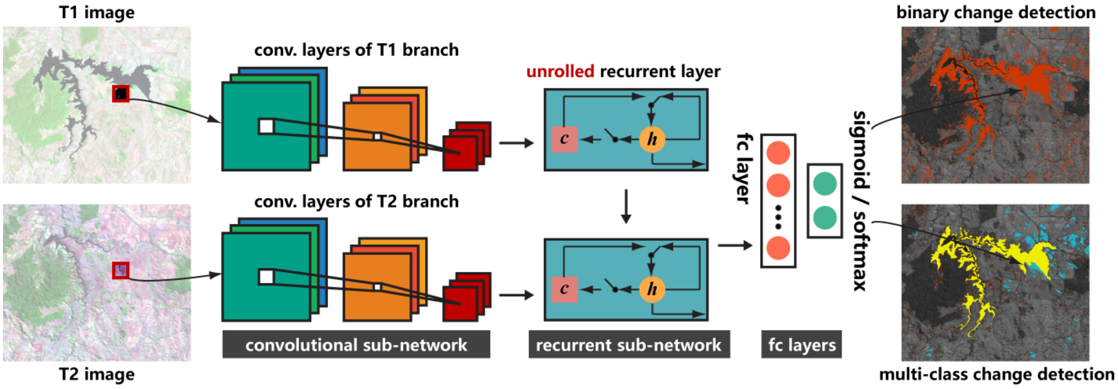


Figure 4.5: Overview of the proposed recurrent convolutional neural network for change detection.

4.3.2 Methodology

The architecture of the proposed recurrent convolutional neural network (ReCNN), as shown in Figure 4.5, is made up of three components, including a convolutional sub-network, a recurrent sub-network, and fully connected layers, from bottom to top.

To acquire a joint spectral-spatial-temporal feature representation for change detection, at the bottom of our network, convolutional layers automatically extract feature maps from each input. On top of the convolutional sub-network, a recurrent sub-network takes the feature representations produced by convolutional layers as inputs to exploit the temporal dependency in the bi-temporal images. The third part is two fully connected layers widely used in classification problems. Although ReCNN is composed of different kinds of network architectures (i.e., CNN, RNN, and fully connected network) it can be trained end-to-end by back-propagation with one loss function, due to the differential properties of all these components.

Let $\boldsymbol{X}^{T_1}$ and $\boldsymbol{X}^{T_2}$ represent a pair of multispectral images acquired over the same geographical area at two different times T_1 and T_2, respectively. Let $\boldsymbol{x}^{T_1}$ and $\boldsymbol{x}^{T_2}$ be two patches taken from the exact same location in two images. $\boldsymbol{y}$ is a label that indicates the category (i.e., changed, unchanged, or change-type) that the pair of patches belongs to. The flowchart of the proposed ReCNN can be summarized as follows:

1. First, the 3D multispectral patch $\boldsymbol{x}^{T_1}$ is fed into T_1 branch of the convolutional sub-network, which transforms it to an abstract feature vector $\boldsymbol{f}^{T_1}$.

2. Then, the recurrent sub-network receives $\boldsymbol{f}^{T_1}$ and calculates the hidden state information for the current input; it also restores that information in the meantime.

3. Subsequently, $\boldsymbol{x}^{T_2}$ is input to the T_2 branch for extracting spectral-spatial feature $\boldsymbol{f}^{T_2}$, it is fed into the recurrent layer simultaneously with the state information of $\boldsymbol{f}^{T_1}$, and the activation at time T_2 is computed by a linear interpolation between existing value and the activation of the previous time T_1.

4. Finally, fully connected layers of the ReCNN predict the label of the input bi-temporal multispectral patches by looping through the entire sequence.

The entire change detection map can be obtained by applying the network to all pixels in the image.

Spectral-Spatial Feature Extraction via the Convolutional Sub-Network. We make use of dilated convolution to construct convolutional layers in the network because, for our task, it is able to offer a slightly better performance than a traditional convolution operation. The dilated convolution [82] was originally designed for the efficient computation of the undecimated wavelet transform in the "algorithme à trous" scheme. This algorithm makes it possible to calculate responses of any layer at any desirable resolution and can be applied post-hoc, once a network has been trained. Let $F : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be a discrete function. Let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and let $k : \Omega_r \rightarrow \mathbb{R}$ be a discrete filter of size $(2r+1)^2$. The traditional discrete convolution operation $*$ can be defined as follows:

$$(F * k)(\mathrm{p}) = \sum_{\mathrm{s}+\mathrm{t}=\mathrm{p}} F(\mathrm{s})k(\mathrm{t})\,. \tag{4.21}$$

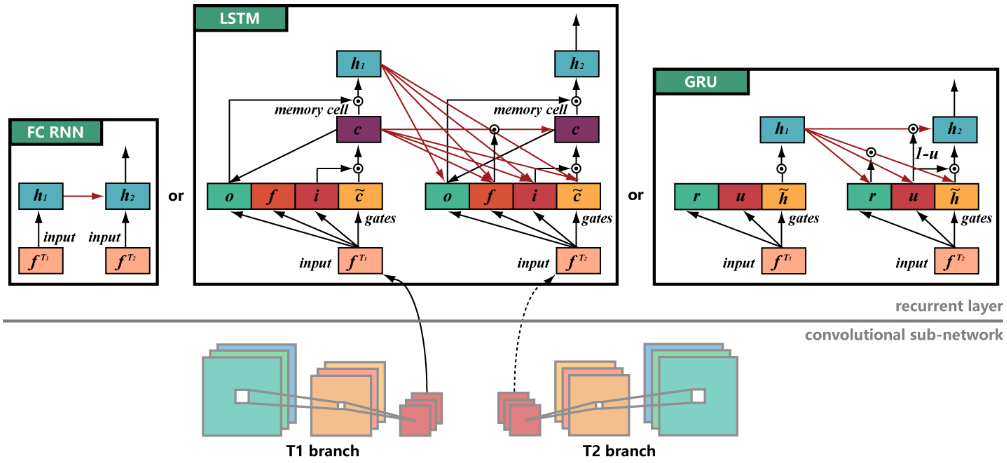


Figure 4.6: Graphic models of fully connected RNN, LSTM, and GRU. In LSTM, $\boldsymbol{o}$, $\boldsymbol{f}$, $\boldsymbol{i}$, $\tilde{\boldsymbol{c}}$, and $\boldsymbol{c}$ are output gates, forget gates, input gates, new memory cell contents, and memory cells, respectively. In GRU, the reset and update gates are denoted by $\boldsymbol{r}$ and $\boldsymbol{u}$, and $\tilde{\boldsymbol{h}}$ and $\boldsymbol{h}$ are the candidate activation and final activation.

This operation can be generalized. Let l be a dilation rate and let $*_l$ be defined as

$$(F *_l k)(\mathrm{p}) = \sum_{\mathrm{s}+l\mathrm{t}=\mathrm{p}} F(\mathrm{s})k(\mathrm{t})\,. \tag{4.22}$$

We will refer to $*_l$ as a dilated convolution or an l-dilated convolution.

The usage of dilated convolution in our network allows us to exponentially enlarge the field of view with linearly increasing number of parameters, providing a significant parameter reduction while increasing the effective field of view. Note that recent studies found that a large field of view actually plays an important role in pattern recognition tasks. This can be easily understood by an analogy that states the fact that humans usually confirm the category of a pixel by referring to its surrounding context region.

Modeling Temporal Dependency by the Recurrent Sub-Network. RNNs can exhibit dynamic temporal behavior, which is in line with our purpose; i.e., modeling temporal dependency between the T_1 and T_2 data. Hence three types of RNN architectures, namely, fully connected RNN, LSTM, and GRU, are used to construct the recurrent sub-network in our network (cf. Figure 4.6).

4.3.3 Results

Dataset. Taizhou dataset consists of two images covering the city of Taizhou, China, in March 2000 and February 2003, with a WGS-84 projection and a coordinate range of 31°14′56N–31°27′39N, 120°02′24E–121°07′45E. These two images both consist of 400 × 400 pixels, and changes between them mainly involve city expansion. The available

Table 4.3: Accuracy Comparison of Binary Change Detection on the Taizhou Dataset.

	Taizhou City			
	OA	Kappa	Unchanged	Changed
CVA	83.82	0.3202	97.38	27.10
PCA	94.63	0.8181	**99.79**	74.51
MAD	94.62	0.8168	98.47	78.52
IRMAD	95.14	0.8313	99.35	77.53
CNN	96.03	0.8667	98.97	83.75
RNN	96.50	0.8884	97.58	91.96
ReCNN-FC	98.35	0.9470	98.94	95.86
ReCNN-GRU	98.67	0.9571	99.23	96.30
ReCNN-LSTM	**98.73**	**0.9592**	99.20	**96.77**

manually annotated samples of this data set for multi-class change detection cover four classes of interest; i.e., unchanged area, city change/expansion (bare soils, grasslands, or cultivated fields to buildings or roads), soil change (cultivated field to bare soil), and water change (non-water regions to water regions).

Quantitative Evaluation. Table 4.3 shows the quantitative comparison with other methods.

For more experimental results and technical details, please refer to Appendix C.

4.4 Perception: Not All Spectral Bands Are Equal

4.4.1 Motivation

The unique asset of hyperspectral images is their rich spectral content in comparison with high-resolution aerial images and natural images in the computer vision field. Although there exist already a number of works that have focused on using CNNs for hyperspectral data classification, we notice that in the community, the following questions have not been well addressed by now.

1. Do all spectral bands contribute equally to *a CNN* for classification tasks?
2. If no, how to *task-drivenly* find informative bands that can help hyperspectral data classification *in an end-to-end network*?
3. Is it possible to improve classification results of a CNN by emphasising informative bands and suppressing less useful ones in the network?

These questions give us an incentive to devise a novel network called spectral attention module-based convolutional network for hyperspectral image classification. Inspired by recent advances in the attention mechanism of networks [21, 118, 119], which enables feature interactions contribute differently to predictions, we design a channel attention mechanism for analyzing the significance of different spectral bands and recalibrating them. More importantly, the significance analysis is automatically learned from tasks and hyperspectral data in an end-to-end network without any human domain knowledge.

4.4.2 Methodology

The spectral attention module in our model transforms a patch $\boldsymbol{x}$ of a hyperspectral image into a new representation $\boldsymbol{z}$ via the following mapping:

$$\boldsymbol{F} : \boldsymbol{x} \rightarrow \boldsymbol{z} \,, \tag{4.23}$$

where $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^{H \times W \times C}$.

Our aim is to strengthen the representational capacity of a spectral-spatial classification network through explicitly modeling the significance of spectral bands. Therefore, we instantiate $\boldsymbol{F}$ as

$$\boldsymbol{z} = \boldsymbol{x} \odot \boldsymbol{g} \,, \tag{4.24}$$

where $\odot$ is a channel-wise multiplication operation, and $\boldsymbol{g} \in \mathbb{R}^{C}$ represents a set of *spectral gates* applied to individual spectral bands of the patch $\boldsymbol{x}$.

The motivation behind Eq. (4.24) is that we wish to make use of a gating mechanism to recalibrate strengths of different spectral bands of the input, i.e., selectively emphasise useful bands and suppress less informative ones, for image classification problems.

Figure 4.7 illustrates the architecture of the spectral attention module-equipped convolutional network.

Now we need a way to aggregate the spectral-spatial information of $\boldsymbol{x}$ across spatial domain to produce a collection of spectral gates $\boldsymbol{g}$.

Convolution operation is an ideal candidate, as 1) it is able to spatially shrink the input patch and 2) its differential property allows end-to-end learning. In general, a convolutional filter operates with a local receptive field (e.g., 3×3 in VGG-16 network), which leads to the fact that the output is not capable of utilizing contextual information outside of this region. This is a severe issue for our case because the spectral gates $\boldsymbol{g}$ in our model are expected to be derived from the whole spectral-spatial information. To tackle this problem, we distill global spatial information into the spectral gates by using global convolution. Formally, let $\boldsymbol{f} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_C]$ denote a set of convolutional filters and their sizes are both $H \times W$, where $\boldsymbol{f}_c$ refers to the c-th filter. Thus the c-th spectral gate g_c can be calculated as follows:

$$g_c = \boldsymbol{x} * \boldsymbol{f}_c = \sum_{i=1}^{C} \boldsymbol{x}_i * \boldsymbol{f}_c^i \,, \tag{4.25}$$

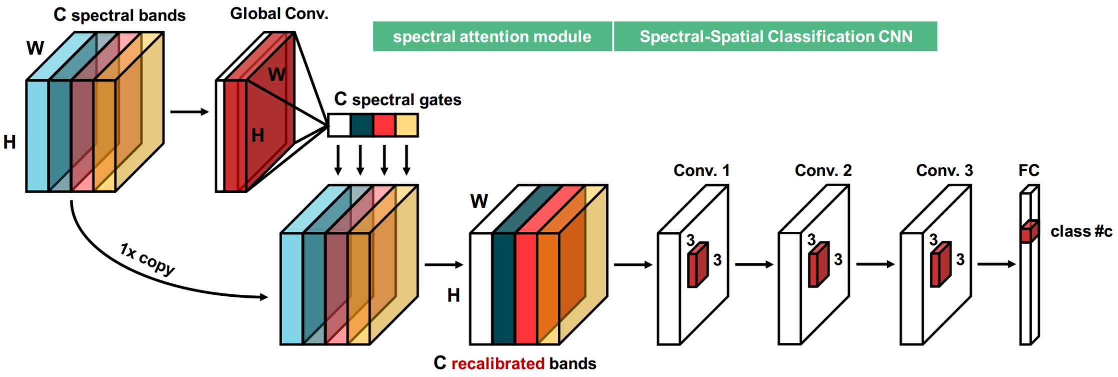


Figure 4.7: Overall architecture of the proposed gating mechanism, spectral attention module, for hyperspectral classification problems. We would like to exploit this module to learn and recalibrate strengths of different spectral bands, i.e., selectively emphasise useful bands and suppress less informative ones, for image classification problems. To this end, we first learn a set of spectral gates by using global convolution and then apply them to individual spectral bands.

where $*$ represents convolution, and $\boldsymbol{f}_c^i$ and $\boldsymbol{x}_i$ are separately the i-th channels of the c-th filter and $\boldsymbol{x}$. Taking into account that the field of view of the global convolution is equal to the spatial size of $\boldsymbol{x}$, g_c is actually calculated by the inner product of $\boldsymbol{x}_i$ and $\boldsymbol{f}_c^i$ (both $\boldsymbol{x}_i$ and $\boldsymbol{f}_c^i$ are vectorized into columns), i.e., Eq. (4.25) can be rewritten as follows:

$$g_c = \sum_{i=1}^{C} \langle \boldsymbol{x}_i, \boldsymbol{f}_c^i \rangle = \sum_{i=1}^{C} \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{f}_c^i \,. \tag{4.26}$$

From Eq. (4.26), the spectral gates $\boldsymbol{g}$ can be considered as a series of global descriptors, which are capable of representing spectral-spatial features of $\boldsymbol{x}$.

Thus according to Eq. (4.24), we can associate the c-th spectral gate g_c with the c-th spectral band of $\boldsymbol{x}$ to obtain the recalibrated $\boldsymbol{z}_c$ via

$$\boldsymbol{z}_c = \boldsymbol{x}_c \sum_{i=1}^{C} \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{f}_c^i \,. \tag{4.27}$$

So far, we can obtain an initial spectral attention module (as shown in Eq. (4.27)), but there still exist three issues we should address:

- Given complex spectral-spatial properties of hyperspectral images, we wish the spectral gates in this module are capable of learning a non-linear mapping, instead of a linear one, from the input.
- The attention module should model a non-mutually-exclusive relationship between spectral bands, as we would like to ensure that multiple bands can be emphasised at the same time (unlike one-hot activation in softmax).

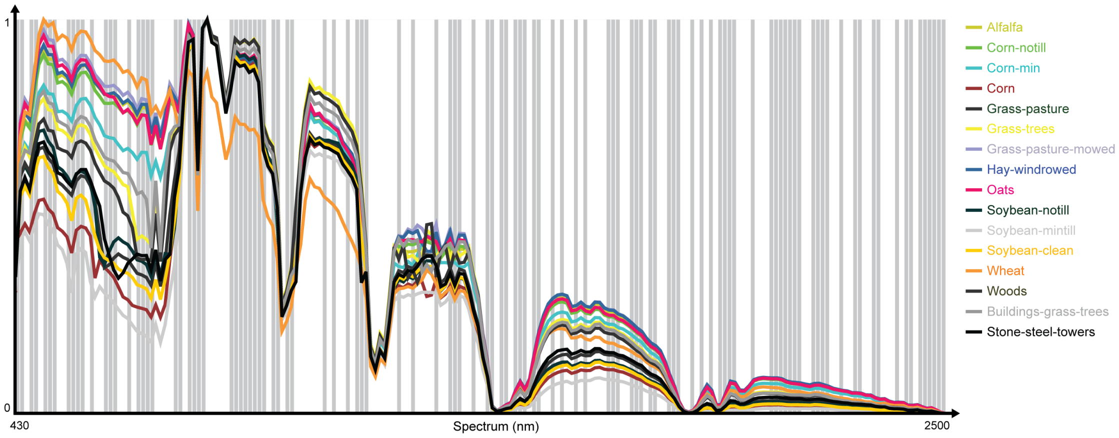


Figure 4.8: Average reflectance spectrum of each class and learned spectral gates on the Indian Pines data set. From this figure, we can observe that the spectral attention module mainly pays attention on spectral bands that provide visual cues to distinguish different categories.

- The gates should be bounded (e.g., between 0 and 1), easily differentiable, and monotonic (good for convex optimization).

To meet these three requirements, we modify spectral gates in the initial spectral attention module as follows:

$$\begin{aligned} g_c =& \frac{1}{1+\exp(-\boldsymbol{x} * \boldsymbol{f}_c)} \\ =& \frac{1}{1+\exp(-\sum_{i=1}^{C} \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{f}_c^i)} . \end{aligned} \tag{4.28}$$

Hence the final version of the spectral attention module can be written as

$$\boldsymbol{z}_c = \boldsymbol{x}_c \frac{1}{1+\exp(-\sum_{i=1}^{C} \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{f}_c^i)} . \tag{4.29}$$

Figure 4.8 shows the learned gates on the Indian Pines dataset. From this figure, we can observe that the spectral attention module mainly pays attention on spectral bands that provide visual cues to distinguish different categories.

4.4.3 Results

Quantitative Evaluation. Table 4.4 shows the quantitative comparison with other methods on the Pavia University dataset. Prior to training models, we normalize each channel of the hyperspectral data to the range between 0 and 1.

For more experimental results and technical details, please refer to Appendix D.

Table 4.4: Accuracy Comparisons for the Pavia University Scene. Bold numbers indicate the best performance.

Class Name	RF	SVM	CCF	SICNN	2D-CNN	SpecAttenNet
Asphalt	81.54	82.37	86.59	84.21	83.85	**86.71**
Meadows	55.39	67.87	72.33	91.10	96.09	**98.47**
Gravel	53.07	69.18	71.75	64.36	**81.47**	77.47
Trees	98.76	98.37	**99.09**	95.53	96.12	96.83
Metal Sheets	99.11	99.41	**99.78**	97.70	98.74	98.81
Bare Soil	79.10	93.64	**97.26**	56.53	49.79	53.11
Bitumen	84.36	91.20	**91.88**	77.29	79.32	77.82
Bricks	91.39	92.59	94.92	**95.57**	88.89	94.43
Shadows	97.47	96.94	**98.73**	96.20	94.19	96.30
OA	71.53	79.89	83.36	85.25	86.93	**89.14**
AA	82.24	87.95	**90.26**	84.28	85.38	86.66
Kappa	0.6504	0.7491	0.7905	0.8041	0.8242	**0.8535**

4.5 Perception: Multitask Learning Network for Vehicle Instance Segment

4.5.1 Motivation

The last decade has witnessed dramatic progress in modern remote sensing technologies – along with the launch of small and cheap commercial high-resolution satellites and the now widespread availability of unmanned aerial vehicles (UAVs) – which facilitates a diversity of applications, such as urban management, monitoring of land changes, and traffic monitoring. Among these applications, object extraction from very high-resolution remote sensing images/videos has gained increasing attention in the remote sensing community in recent years, particularly vehicle extraction, due to successful civil applications. Vehicle extraction, however, is still a challenging task, mainly because it is easily affected by several factors, e.g., vehicle appearance variation, the effects of shadow, illumination, a complicated and cluttered background, etc. Existing vehicle extraction approaches can be roughly divided into two categories: vehicle detection and vehicle semantic segmentation.

The existence of “touching” vehicles in a remote sensing image makes it quite hard for most vehicle semantic segmentation methods to separate objects individually, while in most cases, we need to know not only which pixels belong to vehicles (vehicle semantic segmentation problem) but also the exact number of vehicles (vehicle detection task). This drives us to examine instance-oriented vehicle segmentation. Vehicle instance segmentation seeks to identify the semantic class of each pixel (i.e., vehicle or non-vehicle) as well as associate each pixel with a physical instance of a vehicle. This is contrasted

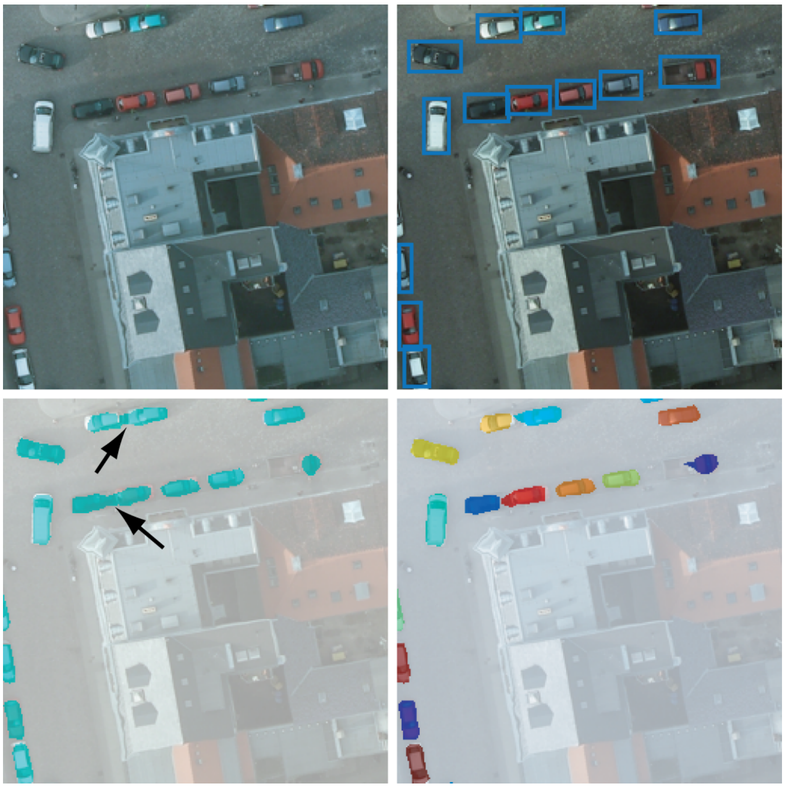


Figure 4.9: An illustration of different vehicle extraction methods. From left to right and top to bottom: input image, vehicle detection, semantic segmentation, and vehicle instance segmentation. The challenge of vehicle instance segmentation is that some vehicles are segmented incorrectly. While most pixels belonging to the category are identified correctly, they are not correctly separated into instances (see arrows in the lower left image).

to vehicle semantic segmentation, which is only concerned with the above-mentioned first task. In this work, we are interested in vehicle instance segmentation in a complex, cluttered, and challenging background from aerial images and videos.

4.5.2 Methodology

We formulate the vehicle instance segmentation task by two subproblems, namely vehicle detection and semantic segmentation. The training set is denoted by $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)\}$, where $i = 1, 2, \cdots, N$ and N is the number of training samples. Since we consider each image independently, the subscript i is dropped hereafter for notational simplicity. $\boldsymbol{x} = \{x_j, j = 1, 2, \cdots, |\boldsymbol{x}|\}$ represents a raw input image, $\boldsymbol{y} = \{y_j, j = 1, 2, \cdots, |\boldsymbol{x}|, y_j \in \{0, 1\}\}$ denotes its corresponding manually annotated pixel-wise segmentation mask, and $\boldsymbol{z} = \{\boldsymbol{r}_k, k = 0, 1, \cdots, K\}$ is the instance label, where $\boldsymbol{r}_k$ indicates a set of pixels inside

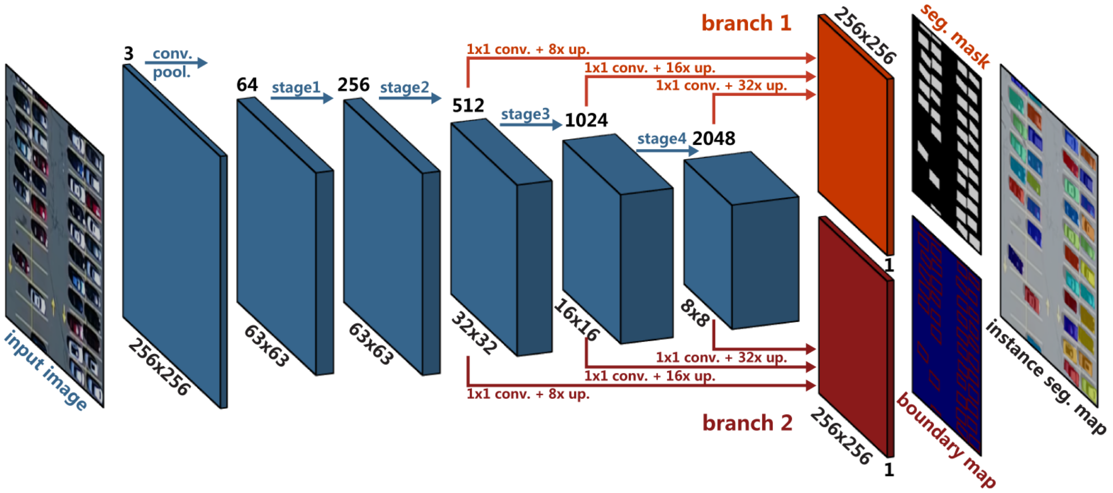


Figure 4.10: Overall architecture of the proposed semantic boundary-aware ResFCN. We propose to use such a unified multi-task learning network for vehicle instance segmentation, which creates two separate, yet identical branches to jointly optimize two complementary tasks, namely, vehicle semantic segmentation and semantic boundary detection. The latter subproblem is beneficial for differentiating "touching" vehicles and further improving the instance segmentation performance.

the k-th region[1]. K is the total number of vehicle instances in the image, and $\boldsymbol{r}_0$ is the background area. When k takes other values, it denotes the corresponding vehicle instance. Note that instance labels only count vehicle instances, thus they are commutative. Our aim is to segment vehicles while ensuring that all instances are differentiated. In this work, we approximate vehicle detection by semantic boundary detection[2]. We generate semantic boundary labels $\boldsymbol{b}$ through $\boldsymbol{z}$ to train a boundary detector, in which $\boldsymbol{b} = \{b_j, j = 1, 2, \cdots, |\boldsymbol{x}|, b_j \in \{0, 1\}\}$ and b_j equals 1 when it belongs to boundaries.

Here we make use of a ResFCN to produce good likelihood maps of vehicles. It is, however, still difficult to differentiate vehicles with a very close distance by only leveraging the probability of vehicles, due to the ambiguity in "touching" regions. This is rooted in the loss of spatial details caused by max-pooling layers (downsampling) along with feature abstraction. The semantic boundaries of vehicles provide good complementary cues that can be used for separating instances.

Some approaches in computer vision and remote sensing have been explored for modeling segmentation and boundary prediction jointly in a combinatorial framework. For example, Kirillov et al. [125] propose InstanceCut, which represents instance segmentation by two modalities, namely a semantic segmentation and all instance-boundaries. The former is computed from a CNN for semantic segmentation, and the latter is derived from

[1] Regions in the image satisfy $\boldsymbol{r}_k \cap \boldsymbol{r}_t = \varnothing, \forall k \neq t$ and $\cup \boldsymbol{r}_k = \Omega$, in where Ω is the whole image region.

[2] Semantic boundary detection is to detect the boundaries of each object instance in the images. Compared to edge detection, it focuses more on the association of boundaries and their object instances.

a instance-aware edge detector. But this approach does not address end-to-end learning. In the remote sensing community, Marmanis et al. [126] propose a two-step model that learns a CNN to separately output edge likelihoods at multiple scales from color-infrared (CIR) and height data. Then, the boundaries detected with each source are added as an extra channel to each source, and a network is trained for semantic segmentation purposes. The intuition behind this work is that using predicted boundaries helps to achieve sharper segmentation maps. In contrast, we train one end-to-end network that takes as input color images and predicts segmentation maps and object boundaries, in order to augment the performance of segmentation at instance level.

To this end, we train a deep semantic boundary-aware ResFCN for effective vehicle instance segmentation (i.e., segmenting the vehicles and splitting clustered instances into individual ones). Fig. 4.10 shows an overview of the proposed network. Specifically, we formulate it as a unified multi-task learning network architecture by exploring the complementary information (i.e., vehicle region and semantic boundaries), instead of treating the vehicle segmentation problem as an independent and single task, which can simultaneously learn the detections of vehicle regions and corresponding semantic boundaries. As shown in Fig. 4.10, the feature representations extracted from multiple residual blocks are upsampled with two separate, yet identical branches to predict the semantic segmentation masks of vehicles and semantic boundaries, respectively. In each branch, the mask is estimated by the ResFCN with multi-level contextual features. Since we have only two categories (foreground/vehicles vs. background and semantic boundaries vs. non-boundaries), sigmoid and binary cross-entropy loss are used to train these two branches. Formally, the network training can be formulated as a pixel-level binary classification problem regarding ground truth segmentation masks, including vehicle instances and semantic boundaries, as shown in the following:

$$\mathcal{L}(x;\boldsymbol{W}) = \mathcal{L}_s(x;\boldsymbol{W}_n,\boldsymbol{W}_s) + \lambda\mathcal{L}_b(x;\boldsymbol{W}_n,\boldsymbol{W}_b)\,, \tag{4.30}$$

where

$$\begin{aligned}\mathcal{L}_s &= -\sum_{x\in\boldsymbol{x}}[y\log\sigma_s(x) + (1-y)\log(1-\sigma_s(x))]\,,\\ \mathcal{L}_b &= -\sum_{x\in\boldsymbol{x}}[b\log\sigma_b(x) + (1-b)\log(1-\sigma_b(x))]\,.\end{aligned} \tag{4.31}$$

$\mathcal{L}_s(x;\boldsymbol{W}_n,\boldsymbol{W}_s)$ and $\mathcal{L}_b(x;\boldsymbol{W}_n,\boldsymbol{W}_s)$ denote losses for estimating vehicle regions and semantic boundaries, respectively. σ indicates the sigmoid function. We train the network using this joint loss, and the final instance segmentation map is produced by the first branch of the network in test phase. Vehicle instances are obtained by computing connected regions in the predicted segmentation map. Inside a region, pixels belong to the same vehicle; while different regions mean different instances. Our motivation is that jointly estimating segmentation and boundary map in a multi-task network with such a joint loss can offer a better segmentation result at instance level for aerial images. Note that we do not make use of any post-processing operations, such as fusing the segmentation and boundary map, as we want to directly evaluate the performance of this network architecture.

Note that the multi-task learning network is optimized in an end-to-end fashion. This joint multi-task training procedure has several merits. First, in the application of vehicle instance segmentation, the multi-task learning network architecture is able to provide complementary semantic boundary information, which is helpful in differentiating the clustered vehicles, improving the instance-level segmentation performance. Second, the discriminative capability of the network's intermediate feature representations can be improved by this architecture because of multiple regularizations on correlated tasks. Therefore, it can increase the robustness of instance segmentation performance.

4.5.3 Results

Dataset. The task of vehicle instance segmentation currently lacks a compelling and challenging benchmark dataset to produce quantitative measurements and to compare with other approaches. While the ISPRS Potsdam dataset has clearly boosted research in semantic segmentation of high-resolution aerial imagery, it is not as challenging as certain practical scenes, such as a busy parking lot, where vehicles are often parked so close that it is quite hard to separate them, particularly from an aerial view. To this end, in this work, we propose our new challenging Busy Parking Lot UAV Video dataset that we built for the vehicle instance segmentation task. The UAV video was acquired by a camera onboard a UAV covering the parking lot of Woburn Mall, in Woburn, Massachusetts, USA. The video comprises 1920 × 1080 pixels with a spatial resolution of about 15 cm per pixel at 24 frames per second and with a length of 60 seconds. We have manually annotated pixel-wise instance segmentation masks for 5 frames (at 1, 15, 30, 45, and 59 seconds); i.e., the annotation is dense in space and sparse in time to allow for the evaluation of methods with this long sequence. The Busy Parking Lot dataset is challenging because it presents a high range of variations, with a diversity of vehicle colors, effects of shadow, several slightly blurred regions, and vehicles that are parked too close. We train networks on the ISPRS Potsdam dataset and then perform vehicle instance segmentation using the trained networks on this video dataset.

Quantitative Evaluation. Table 4.3 shows the quantitative comparison with other methods.

For more experimental results and technical details, please refer to Appendix E.

Table 4.5: Segmentation Results of Different Methods on Busy Parking Lot UAV Video Dataset (Instance-level Dice Similarity Coefficient)

Model	1s	15s	30s	45s	59s
Inception-FCN	26.81	26.06	25.68	22.89	23.77
B-Inception-FCN	32.37	33.07	33.34	30.44	31.26
Xception-FCN	72.74	72.74	72.85	72.47	71.31
B-Xception-FCN	77.31	**77.50**	77.22	77.13	76.32
ResFCN	71.17	71.47	71.76	68.82	72.73
B-ResFCN	**78.84**	77.33	**79.13**	**77.83**	**79.39**

4.6 Reasoning: Relational Reasoning in Networks for Semantic Segmentation

4.6.1 Motivation

Although with more complicated and deeper networks and more labeled samples, there is a technical hurdle in the application of CNNs to semantic image segmentation—modeling contextual information.

It has been well recognized in the computer vision community for years that contextual information, or *relation*, is capable of offering important cues for semantic segmentation tasks. For instance, spatial relations can be considered semantic similarity relationships among regions in an image. In addition, spatial relations also involve compatibility and incompatibility relationships, i.e., a vehicle is likely to be driven or parked on pavements. Unfortunately, only convolution layers cannot model such spatial relations due to their local valid receptive field[3].

Nevertheless, under some circumstances, spatial relations are of paramount importance, particularly when a region in an image exhibits significant visual ambiguities. To address this issue, several attempts have been made to introduce spatial relations into networks by using either graphical models or spatial propagation networks. However, these methods seek to capture global spatial relations implicitly with a chain propagation way, whose effectiveness depends heavily on the learning effect of long-term memorization. Consequently, these models may not work well in some cases like aerial scenes, in which long-range spatial relations often exist (cf. Figure 4.11). Hence, explicit modeling of long-range relations may provide additional crucial information but still remains underexplored for semantic segmentation.

[3] Feature maps from deep CNNs like ResNet usually have large receptive fields due to deep architectures, whereas the study of [127] has shown that CNNs are apt to extract information mainly from smaller regions in receptive fields, which are called valid receptive fields.

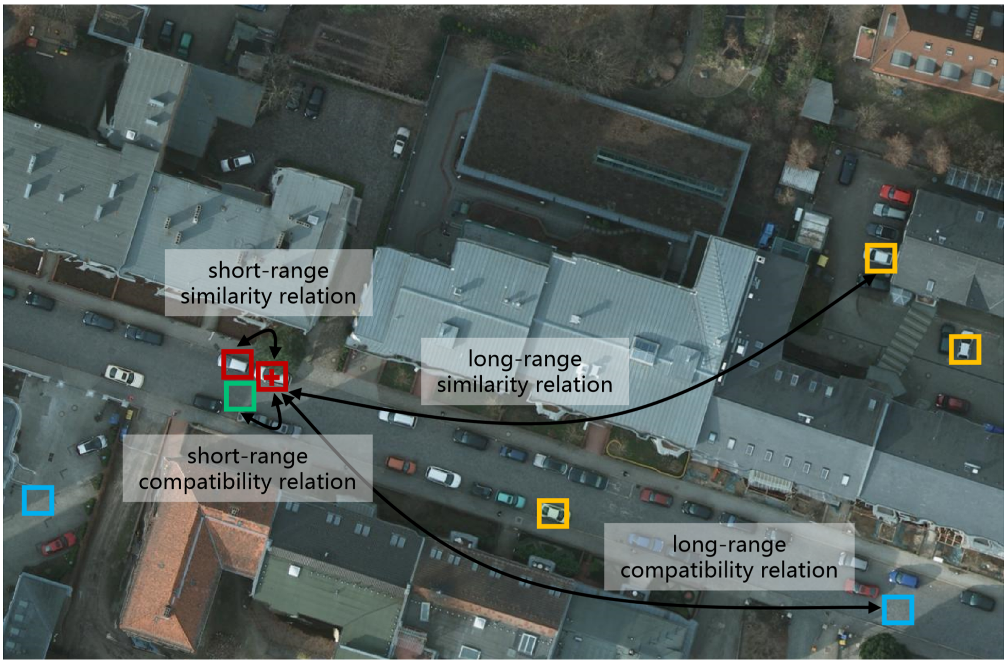

short-range similarity relation
long-range similarity relation
short-range compatibility relation
long-range compatibility relation

Figure 4.11: Illustration of long-range spatial relations in an aerial image. Appearance similarity or semantic compatibility between patches within a local region (red–red and red–green) and patches in remote regions (red–yellow and red–blue) underlines our global relation modeling.

4.6.2 Methodology

The proposed network takes VGG-16 as a backbone to extract multi-level features. Outputs of the third, forth, and fifth layers are fed into the channel and spatial relation modules for generating relation-augmented features. These features are subsequently fed into respective convolutional layers with 1×1 filters to squash the number of channels to the number of categories. Finally, the convolved feature maps are upsampled to a desired full resolution and element-wise added to generate final segmentation maps.

Spatial Relation Module. In order to capture global spatial relations, we employ a spatial relation module, where the spatial relation is defined as a composite function with the following equation:

$$\mathrm{SR}(\boldsymbol{x}_i, \boldsymbol{x}_j) = f_{\phi_s}(g_{\theta_s}(\boldsymbol{x}_i, \boldsymbol{x}_j))\,. \tag{4.32}$$

Denote by $\boldsymbol{X} \in \mathbb{R}^{C\times H\times W}$ a random variable representing a set of feature maps. $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two feature-map vectors, identified by spatial positions indices i and j. The size of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is $C \times 1 \times 1$. To model a compact relationship between them, we make use of an embedding dot product as g_{θ_s} instead of a multilayer perceptron (MLP), and the latter is commonly used in relational reasoning modules [120, 122]. Particularly, g_{θ_s} is

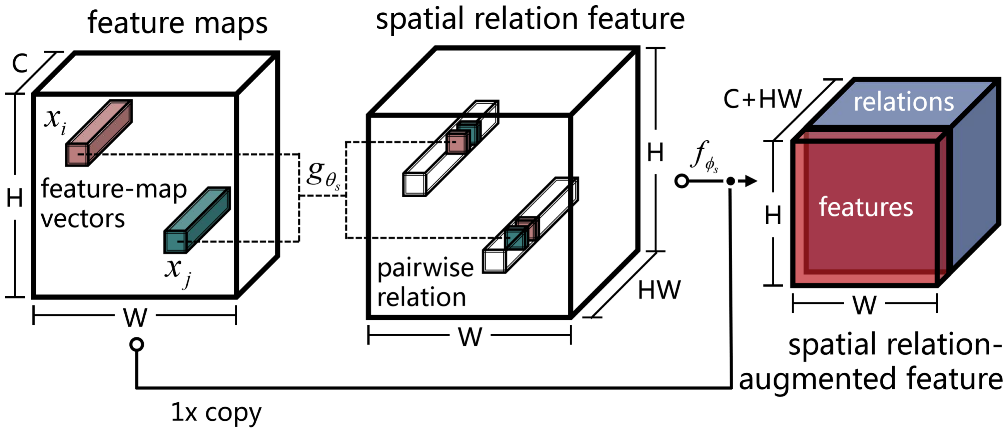


Figure 4.12: Diagrams of spatial relation module.

defined as follows:

$$g_{\theta_s}(\boldsymbol{x}_i, \boldsymbol{x}_j) = u_s(\boldsymbol{x}_i)^T v_s(\boldsymbol{x}_j)\,, \tag{4.33}$$

where $u_s(\boldsymbol{x}_i) = \boldsymbol{W}_{u_s}\boldsymbol{x}_i$ and $v_s(\boldsymbol{x}_j) = \boldsymbol{W}_{v_s}\boldsymbol{x}_j$. $\boldsymbol{W}_{u_s}$ and $\boldsymbol{W}_{v_s}$ are weight matrices and can be learned during the training phase. Considering computational efficiency, we realize Eq. (4.33) in matrix format with the following steps:

1. Feature maps $\boldsymbol{X}$ are fed into two convolutional layers with 1×1 filters to generate $u_s(\boldsymbol{X})$ and $v_s(\boldsymbol{X})$, respectively.
2. Then $u_s(\boldsymbol{X})$ and $v_s(\boldsymbol{X})$ are reshaped (and transposed) into $HW \times C$ and $C \times HW$, correspondingly.
3. Eventually, the matrix multiplication of $u_s(\boldsymbol{X})$ and $v_s(\boldsymbol{X})$ is conducted to produce a $HW \times HW$ matrix, which is further reshaped to form a spatial relation feature of size $HW \times H \times W$.

It is worth nothing that the spatial relation feature is not further synthesized (, summed up), as fine-grained contextual characteristics are essential in semantic segmentation tasks. Afterwards, we select the ReLU function as f_{ϕ_s} to eliminate negative spatial relations.

However, relying barely on spatial relations leads to a partial judgment. Therefore, we further blend the spatial relation feature and original feature maps $\boldsymbol{X}$ as follows:

$$\boldsymbol{X}_s = [\boldsymbol{X}, \mathrm{SR}(\boldsymbol{X})]\,. \tag{4.34}$$

Here we simply use a concatenation operation, i.e., $[\cdot,\cdot]$, to enhance original features with spatial relations. By doing so, output features are abundant in global spatial relations, while high-level semantic features are also preserved.

Figure 4.12 shows the diagrams of the spatial relation module.

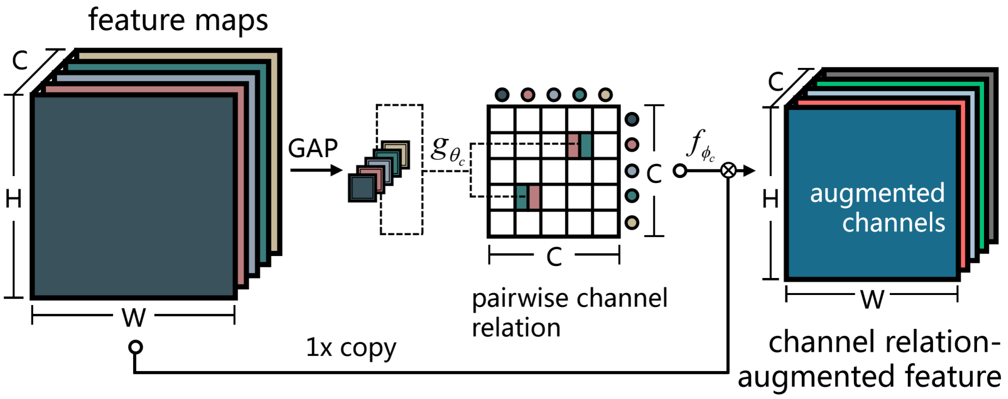


Figure 4.13: Diagrams of channel relation module.

Channel Relation Module. Although the spatial relation module is capable of capturing global contextual dependencies for identifying various objects, misdiagnoses happen when objects share similar distribution patterns but vary in channel dimensionality. In addition, a recent work [21] has shown the benefit of enhancing channel encoding in a CNN for image classification tasks. Therefore, we propose a channel relation module to model channel relations, which can be used to enhance feature discriminabilities in the channel domain. Similar to the spatial relation module, we define the channel relation as a composite function with the following equation:

$$\mathrm{CR}(\boldsymbol{X}_p, \boldsymbol{X}_q) = f_{\phi_c}(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q))\,, \tag{4.35}$$

where the input is a set of feature maps $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_C\}$, and $\boldsymbol{X}_p$ as well as $\boldsymbol{X}_q$ represents the p-th and the q-th channels of $\boldsymbol{X}$. Embedding dot product is employed to be g_{θ_c}, defined as

$$g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q) = u_c(\mathrm{GAP}(\boldsymbol{X}_p))^T v_c(\mathrm{GAP}(\boldsymbol{X}_q))\,, \tag{4.36}$$

for capturing global relationships between feature map pairs, where GAP$(\cdot)$ denotes the global average pooling function. Notably, considering that the preservation of spatial structural information distracts the analysis of channel inter-dependencies, we adopt averages of $\boldsymbol{X}_p$ and $\boldsymbol{X}_q$ as channel descriptors before performing dot product. More specifically, we feed feature maps into a global average pooling layer for generating a set of channel descriptors of size $C \times 1 \times 1$, and then exploit two convolutional layers with 1×1 filters to produce $u_c(\boldsymbol{X})$ and $v_c(\boldsymbol{X})$, respectively. Afterwards, an outer product is performed to generate a $C \times C$ channel relation feature, where the element located at (p, q) indicates $g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q)$.

Furthermore, we emphasize class-relevant channel relations as well as suppress irrelevant channel dependencies by adopting a softmax function as f_{ϕ_c}, formulated as

$$f_{\phi_c}(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q)) = \frac{\exp(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q))}{\sum_{q=1}^{C} \exp(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q))}\,, \tag{4.37}$$

where we take $\boldsymbol{X}_p$ as an example. Consequently, a discriminative channel relation map $\mathrm{CR}(\boldsymbol{X})$ can be obtained, where each element represents the corresponding pairwise channel relation.

To integrate $\mathrm{CR}(\boldsymbol{X})$ and original feature maps $\boldsymbol{X}$, we reshape $\boldsymbol{X}$ into a matrix of $C \times HW$ and employ a matrix multiplication as follows:

$$\boldsymbol{X}_c = \boldsymbol{X}^T \mathrm{CR}(\boldsymbol{X}) \,. \tag{4.38}$$

With this design, the input features are enhanced with channel relations and embedded with not only initial discriminative channel properties but also global inter-channel correlations. Eventually, $\boldsymbol{X}_c$ is reshaped to $C \times H \times W$ and fed into subsequent procedures.

Integration of Relation Modules. In order to jointly enjoy benefits from spatial and channel relation modules, we further aggregate features $\boldsymbol{X}_s$ and $\boldsymbol{X}_c$ to generate spatial and channel relation-augmented features. We investigate two integration patterns, namely serial integration and parallel integration, to blend $\boldsymbol{X}_s$ and $\boldsymbol{X}_c$. For the former, we append the spatial relation module to the channel relation module and infer $\boldsymbol{X}_s$ from $\boldsymbol{X}_c$ instead of $\boldsymbol{X}$, as presented in Eq. (4.32) and Eq. (4.38). For the latter, spatial relation-augmented features and channel relation-augmented features are obtained simultaneously and then aggregated by performing concatenation.

Figure 4.13 shows the diagrams of the channel relation module.

4.6.3 Results

Dataset. Vaihingen dataset[4] is composed of 33 aerial images collected over a 1.38 km^2 area of the city, Vaihingen, with a spatial resolution of 9 cm. The average size of each image is 2494×2064 pixels, and each of them has three bands, corresponding to near infrared (NIR), red (R), and green (G) wavelengths. Notably, DSMs, which indicate the height of all object surfaces in an image, are also provided as complementary data. Among these images, 16 of them are manually annotated with pixel-wise labels, and each pixel is classified into one of six land cover classes. We select 11 images for training, and the remaining five images (image IDs: 11, 15, 28, 30, 34) are used to test our model.

Quantitative Evaluation. Table 4.1 shows the quantitative comparison with other methods.

For more experimental results and technical details, please refer to Appendix F.

[4] `http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html`

Table 4.6: Experimental Results on the Vaihingen Dataset

Model Name	Imp. surf.	Build.	Low veg.	Tree	Car	mean F_1	
SVL+CRF	86.10	90.90	77.60	84.90	59.90	79.90	84.70
RF+dCRF	86.90	92.00	78.3	86.90	29.00	74.60	85.90
CNN-FPL	-	-	-	-	-	83.58	87.83
FCN	88.67	92.83	76.32	86.67	74.21	83.74	86.51
FCN-dCRF	88.80	92.99	76.58	86.78	71.75	83.38	86.65
SCNN	88.21	91.80	77.17	87.23	78.60	84.40	86.43
Dilated FCN	90.19	94.49	77.69	87.24	76.77	85.28	87.70
FCN-FR	**91.69**	**95.24**	79.44	88.12	78.42	86.58	88.92
PSPNet	89.92	94.36	78.19	87.12	72.97	84.51	87.62
RotEqNet	89.50	94.80	77.50	86.50	72.60	84.18	87.50
RA-FCN-srm	91.01	94.86	80.01	88.74	87.16	88.36	89.03
P-RA-FCN	91.46	95.02	80.40	88.56	**87.08**	88.50	89.18
S-RA-FCN	91.47	94.97	**80.63**	**88.57**	87.05	**88.54**	**89.23**

5 Conclusion

5.1 Summary

This thesis explores several deep networks for tackling common perception tasks in the remote sensing community, such as hyperspectral image classification and semantic segmentation of high resolution aerial images. More specifically,

for hyperspectral image analysis,

- a modified GRU-based RNN model was proposed to analyze hyperspectral pixels form a novel perspective, i.e., taking them as sequential data instead of vectors in most existing machine learning methods;
- an end-to-end trainable residual conv-deconv network was proposed for unsupervised spectral-spatial feature learning, and an interesting finding is some learned filters in the *first* convolutional block in such unsupervised learning networks have high-level semantics, which makes "free" object detection possible in hyperspectral images;
- a recurrent convolutional neural network was presented to learn a joint spectral-spatial-temporal feature representation for sequence analysis tasks in remote sensing like change detection;
- for studying if different bands contribute unequally, a spectral attentional module was devised, and it can improve the performance of a CNN by recalibrating spectral bands according to learned attentions to them;

and for the instance segmentation of vehicles in high resolution aerial images and videos,

- a unified multitask learning CNN was proposed to jointly two complementary subtasks, i.e., segmenting vehicle masks and estimating vehicle boundaries.

From perception to reasoning,

- this thesis explores a relational reasoning module-equipped FCN, where both spatial and channel relations are reasoned about for information propagation in aerial image semantic segmentation tasks.

These studies address the questions raised in Section 1.1. For the first question, i.e., what do rich spectra of hyper/multi-spectral imagery bring for deep networks, the findings are:

- On one hand, we can benefit from the rich spectral information offered by hyper-multi-spectral images. For example, some filters learned in the first convolutional block of an unsupervised conv-deconv network have high-level or object-level semantics, which is different from the knowledge in computer vision, where we believe lower layers of a CNN can only learn fairly simple conceptions, e.g., lines with different directions, parts of object boundaries, and various color patterns, and only deeper layers can learn high-level semantics. This special property makes it possible to learn "free" object-specific detectors using hyper/multi-spectral data.
- On the other hand, for a specific task, we do not need all spectral bands. In other words, not all spectral bands contribute equally to a CNN for classification tasks for example. CNNs usually pay more attention to a part of spectral bands (which we call informative bands) and ignore others. And according to experiments, networks tend to select spectral bands with high information entropy. This is in line with studies in hyperspectral band selection, in which information entropy is an important measurement.
- Finally RNN is a good model for processing spectral sequence and time sequence data. But recently, self-attention models show better performance than RNN models in many fields. For instance, in NLP, there is even a trend replacing RNNs with self-attention models. For multi-temporal data analysis in the remote sensing community, it is not clear at the moment. More studies in this direction are expected.

For the second question, i.e., object detection and semantic segmentation in high resolution aerial images, I have the following opinions:

- Touching objects, e.g., close cars in a busy parking lot, are one of hard situations in instance segmentation of remote sensing images. A multi-task learning-based segmentation network is an effective solution. But its inference time is a little bit slow, which cannot meet the requirement of real-time applications. A rotative object detection framework may be an efficient solution, particularly for objects with regular shapes (e.g., cars), by predicting rotative bounding boxes to perfectly fit shapes of objects. This method is much faster than semantic segmentation-based approaches.
- Modeling both short- and long-range relations or dependencies are important for semantic segmentation in remote sensing images. While most existing solutions, e.g., introducing graphic models into networks, do not work well in aerial images. Relational reasoning and self-attention models seem promising solutions because they can explicitly model global relations in images. But one drawback of these methods is they need huge computation and memory costs. How to significantly reduce such costs is important for applying these models to practical, large-scale applications.

For the third question, i.e., can remote sensing data analysis benefit from reasoning learning, I have explored in the sixth work in this thesis, and for more discussions refer to the next section.

5.2 Outlook

Over the past years deep networks have brought a real revolution in remote sensing, producing stunning results in a variety of different applications. For instance, deep network-based remote sensing image classification, object detection, and semantic segmentation systems can now be trained to recognize hundreds of different land cover and land use categories, which sometimes are difficult to distinguish even for humans. Albeit these are indeed impressive advancements, there is no doubt that many problems that are really at the core of AI for Earth observation are far from being solved. This is particularly true for those tasks that involve reasoning, such as induction, deduction, spatial and temporal reasoning, and structure inference. Here I give three examples: spatial reasoning in remote sensing images, temporal reasoning for understanding aerial video data, and multimodal fusion network architecture reasoning.

- So far the automatic interpretation of high-resolution aerial and satellite images has mainly focused on identifying land cover/land use and objects in images – learning to predict their presence (i.e., image classification) and spatial locations (i.e., object detection or semantic segmentation). These object-centric methods have matured significantly in recent years, and most of these gains are a result of making use of deep learning techniques, such as CNNs and RNNs. However, unlike humans, current vision systems for high-resolution remote sensing image analysis represent images as collections of semantic objects and fail to reason about complex spatial relations among them which may be essential for visual understanding. An example is to accurately search "buildings" near a "river". Most current remote sensing image retrieval systems simply search for images containing both "building" and "river" while ignoring their spatial relations (in this case "near"). Therefore, a critical question now is: how do we incorporate both spatial and semantic reasoning effectively to build next-generation intelligent vision systems for Earth observation?
- Along with the launch of small and cheap commercial satellites (e.g., the SkySat-constellation of Terra Bella) and the widespread availability of unmanned aerial vehicles (UAVs), space-borne and high-resolution airborne videos are now accessible at a reasonable cost. In comparison with static images, the unique asset of these remotely sensed videos is their rich temporal content, which can be exploited for a wide range of dynamic Earth observation problems (e.g., human activity recognition, event detection, and real-time traffic and disaster monitoring). However, so far the automatic parsing of such videos has received scant attention in the remote sensing community and still remains underexplored. On the other hand, understanding such complex video data usually needs long-range temporal reason-

ing, which allows a model to analyze the current situation relative to the past and formulate hypotheses on what may happen next. Unfortunately, most existing deep learning models either lack the capability of modeling long-range temporal relations, e.g., CNNs, or capture these relations implicitly in a chain propagation way, whose effectiveness depends heavily on the learning effect of long-term memorization, e.g., RNNs. Hence, enabling explicit temporal reasoning in networks may provide additional crucial information for such video analysis tasks.

- Earth observation data are often multimodal, for example, from optical (multi- and hyperspectral) to Lidar and from synthetic aperture radar (SAR) to video data, where their imaging geometry and content are completely different. Data and information fusion using these complementary data sources in a synergistic way for Earth observation tasks is conceptually of high interest and has been an active research topic in recent years. A crucial problem in data fusion is to develop appropriate architectures to jointly extract information from multimodal data taken from different perspectives and even different imaging modalities. Existing deep learning models for remote sensing data fusion depend upon hand-designed architectures, e.g., Siamese network and multi-stream architecture. However, fusing modalities using these hand-crafted network architectures is not necessarily the most optimal way. Therefore, an important research question is: can optimal multimodal network architectures be automatically reasoned about for the purpose of data fusion?

The study of spatial reasoning in remote sensing images could benefit to a wide range of applications, to name a few, a smarter image retrieval system that can understand both semantics and spatial relations among objects, image captioning, and VQA system for remote sensing data. On the other hand, explorations in temporal reasoning can benefit for tasks like dynamic event recognition and reasoning in UAV videos, anomaly detection in time series data, and future prediction using remote sensing data.

Uncertainty is one of the most common challenges among different areas of remote sensing and still remains under-explored. Everyday voluminous data are being produced by various remote sensing sensors and applications. However, one needs quantitative uncertainty information associated with the data to extract information and distill knowledge from the data. Uncertainty quantification is also a critical scientific effort for both data producers and end users, as the process will provide revealing error characteristics to guide further improvements in data production and rational use of the data. On the other hand, quantifying uncertainty associated with predictions of deep networks is critical for their deployment and use in practical remote sensing applications. For instance, in land use/cover classification and change detection, deep networks have to be able not only to predict accurately, but also to quantify how certain they are with respect to predictions. Hence uncertainty estimation in deep networks for remote sensing applications is very important.

Furthermore, the combination of prior knowledge and deep networks would also be a promising direction. Deep networks are capable of learning powerful feature representations, and these models can be trained in a fully integrated way. However, training deep networks is difficult when we face the small sample size problem (which is commonly seen in remote sensing applications). Injecting prior knowledge into the networks is a principled way to significantly reduce the amount of required training instances, as the models do not need to induce the knowledge from the data itself. For example, a possible way to effectively train deep networks with limited data samples is to encode some structures in network architectures: these structures are able to encode some available domain prior knowledge in networks without relying on massive amount of data to extrapolate it.

Bibliography

[1] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2094–2107, 2014.

[2] Y. Chen, X. Zhao, and X. Jia. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392, 2015.

[3] C. Tao, H. Pan, Y. Li, and Z. Zou. Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2438–2442, 2015.

[4] R. Guo, W. Wang, and H. Qi. Hyperspectral image unmixing using autoencoder cascade. In *Workshop on Hyperspectral Image and Signal Processing (WHISPERS)*, 2015.

[5] Y. Sun, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty. DAEN: Deep autoencoder networks for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2018.2890633.

[6] D. Marinelli, F. Bovolo, and L. Bruzzone. A novel change detection method for multitemporal hyperspectral images based on binary hyperspectral change vectors. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[7] A. Song, J. Choi, Y. Han, and Y. Kim. Change detection in hyperspectral images using recurrent 3D fully convolutional networks. *Remote Sensing*, 10(11):1827, 2018.

[8] Q. Wang, Z. Yuan, Q. Du, and X. Li. Getnet: A general end-to-end 2-D CNN framework for hyperspectral image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, (99):1–11, 2018.

[9] F. Huang, Y. Yu, and T. Feng. Hyperspectral remote sensing image change detection based on tensor and deep learning. *Journal of Visual Communication and Image Representation*, 58:233–244, 2019.

[10] Z. Yuan, Q. Wang, and X. Li. Robust PCANet for hyperspectral image change detection. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.

[11] T. Lei, Q. Zhang, D. Xue, T. Chen, H. Meng, and A. Nandi. End-to-end change detection using a symmetric fully convolutional network for landslide mapping. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[12] H. Zhang, M. Gong, P. Zhang, L. Su, and J. Shi. Feature-level change detection using deep representation and feature change analysis for multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 13(11):1666–1670, 2016.

[13] M. Gong, X. Niu, P. Zhang, and Z. Li. Generative adversarial networks for change detection in multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2310–2314, 2017.

[14] W. Zhao, Z. Wang, M. Gong, and J. Liu. Discriminative feature learning for unsupervised change detection in heterogeneous images based on a coupled neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7066–7080, 2017.

[15] N. Ma, Y. Peng, S. Wang, and D. Liu. Hyperspectral image anomaly targets detection with online deep learning. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2018.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] G. Huang, Z. Liu, L. V. D. Maaten, and K. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[23] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015, 2015.

[24] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015.

[25] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

[26] Z. Zhong, J. Li, Z. Luo, and M. A. Chapman. Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):847–858, 2018.

[27] L. Mou, P. Ghamisi, and X. X. Zhu. Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1):391–406, 2018.

[28] W. Song, S. Li, L. Fang, and T. Lu. Hyperspectral image classification with deep feature fusion network. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3173–3184, 2018.

[29] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza. Deep&dense convolutional neural network for hyperspectral image classification. *Remote Sensing*, 10(9):1454, 2018.

[30] C. Zhang, G. Li, S. Du, W. Tan, and F. Gao. Three-dimensional densely connected convolutional network for hyperspectral remote sensing image classification. *Journal of Applied Remote Sensing*, 13(1):016519, 2019.

[31] B. Fang, Y. Li, H. Zhang, and J. C. Chan. Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism. *Remote Sensing*, 11(2):159, 2019.

[32] K. Hegde, R. Agrawal, Y. Yao, and C. Fletcher. Morph: Flexible acceleration for 3D CNN-based video understanding. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2018.

[33] Y. Wang, Y. Wang, H. Li, C. Shi, and X. Li. Systolic Cube: A spatial 3D CNN accelerator architecture for low power video analysis. In *Design Automation Conference*, 2019.

[34] J. Li, H. Zhang, W. Wan, and J. Sun. Two-class 3D-CNN classifiers combination for video copy detection. *Multimedia Tools and Applications*, pages 1–13, 2018.

[35] J. Stroud, D. Ross, C. Sun, J. Deng, and R. Sukthankar. D3D: Distilled 3D networks for video action recognition. *arXiv:1812.08249*, 2018.

[36] G. Lin, Y. Zhang, G. Xu, and Q. Zhang. Smoke detection on video sequences using 3D convolutional neural networks. *Fire Technology*, pages 1–21, 2019.

[37] H. Qi, J. Li, Q. Wu, W. Wan, and J. Sun. A 3D-CNN based video hashing method. In *International Conference on Digital Image Processing (ICDIP)*, 2018.

[38] C. Li, L. Zhu, D. Zhu, J. Chen, Z. Pan, X. Li, and B. B. Wang. End-to-end multiplayer violence detection based on deep 3D CNN. In *International Conference on Network, Communication and Computing*, 2018.

[39] A. Diba, A. Pazandeh, and L. V. Gool. Efficient two-stream motion and appearance 3D CNNs for video classification. *arXiv:1608.08851*, 2016.

[40] X. Wang, L. Gao, J. Song, and H. Shen. Beyond frame-level cnn: saliency-aware 3-d cnn with lstm for video action recognition. *IEEE Signal Processing Letters*, 24(4):510–514, 2016.

[41] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[42] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In *ACM International Conference on Multimodal Interaction*, 2016.

[43] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] A. Diba, M. Fayyaz, V. Sharma, A. Karami, M. Arzani, R. Yousefzadeh, and L. V. Gool. Temporal 3D ConvNets: New architecture and transfer learning for video classification. *arXiv:1711.08200*, 2017.

[45] Y. Li, H. Zhang, and Q. Shen. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sensing*, 9(1):67, 2017.

[46] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[47] W. Zhao and S. Du. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4544–4554, 2016.

[48] P. Ghamisi, Y. Chen, and X. X. Zhu. A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Geoscience and Remote Sensing Letters*, 13(10):1537–1541, 2016.

[49] W. Li, G. Wu, F. Zhang, and Q. Du. Hyperspectral image classification using deep pixel-pair features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):844–853, 2017.

[50] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362, 2016.

[51] L. Mou, P. Ghamisi, and X. X. Zhu. Unsupervised spectral–spatial feature learning via deep residual Conv-Deconv network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1):391–406, 2018.

[52] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza. Deep&Dense convolutional neural network for hyperspectral image classification. *Remote Sensing*, 10(9):1454, 2018.

[53] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla. Deep pyramidal residual networks for spectral-spatial hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, DOI:10.1109/TGRS.2018.2860125.

[54] F. I. Alam, J. Zhou, A. W.-C. Liew, X. Jia, J. Chanussot, and Y. Gao. Conditional random field and deep feature learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1612–1628, 2019.

[55] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao. Active transfer learning network: A unified deep joint spectral-spatial feature learning model for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1741–1754, 2019.

[56] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[57] K. Cho, B. B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[58] K. Li, H. Xu, Y. Wang, D. Povey, and S. Khudanpur. Recurrent neural network language model adaptation for conversational speech recognition. 2018.

[59] D. Yogatama, Y. Miao, G. Melis, W. Ling, A. Kuncoro, C. Dyer, and P. Blunsom. Memory architectures in recurrent neural network language models. 2018.

[60] S. Heinrich and S. Wermter. Interactive natural language acquisition in a multimodal recurrent neural architecture. *Connection Science*, 30(1):99–133, 2018.

[61] M. Plappert, C. Mandery, and T. Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109:13–26, 2018.

[62] R. Ghaeini, X. Fern, and P. Tadepalli. Interpreting recurrent and attention-based neural models: A case study on natural language inference. *arXiv:1808.03894*, 2018.

[63] D. Hall, D. Klein, D. Roth, L. Gillick, A. Maas, and S. Wegmann. Sequence to sequence transformations for speech synthesis via recurrent neural networks, 2018.

[64] W. Chan, N. Jaitly, Q. Le, O. Vinyals, and N. Shazeer. Speech recognition with attention-based recurrent neural networks, 2018.

[65] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *Ieee Computational intelligence Magazine*, 13(3):55–75, 2018.

[66] M. Morchid. Parsimonious memory unit for recurrent neural networks with application to natural language processing. *Neurocomputing*, 314:48–64, 2018.

[67] L. Deng and Y. Liu. *Deep Learning in Natural Language Processing.* 2018.

[68] L. Kong, G. Melis, W. Ling, L. Yu, and D. Yogatama. Variational smoothing in recurrent neural network language models. *arXiv:1901.09296*, 2019.

[69] A. Jaech, L. Heck, and M. Ostendorf. Domain adaptation of recurrent neural networks for natural language understanding. *arXiv:1604.00117*, 2016.

[70] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen. Recurrent neural network-based sentence encoder with gated attention for natural language inference. *arXiv:1708.01353*, 2017.

[71] W. Yin, K. Kann, M. Yu, and H. Schütze. Comparative study of CNN and RNN for natural language processing. *arXiv:1702.01923*, 2017.

[72] H. Lyu, H. Lu, and L. Mou. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sensing*, 8(6):506, 2016.

[73] L. Mou, P. Ghamisi, and X. X. Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.

[74] M. Rußwurm and M. Körner. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[75] L. Mou, L. Bruzzone, and X. X. Zhu. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):924–935, 2019.

[76] M. Rußwurm and M. Körner. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS International Journal of Geo-Information*, 7(4):129, 2018.

[77] D. Tuia, B. Kellenberger, A. Pérez-Suay, and G. Camps-Valls. A deep network approach to multitemporal cloud detection. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.

[78] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou. Spatial sequential recurrent neural network for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11):4141–4155, 2018.

[79] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[80] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv:1505.07293*, 2015.

[81] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015.

[82] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[83] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[84] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv:1704.06857*, 2017.

[85] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[86] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[87] Y. Guo, Y. Liu, T. Georgiou, and M. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(2):87–93, 2018.

[88] J. Linmans, J. Winkens, B. Veeling, T. Cohen, and M. Welling. Sample efficient semantic segmentation using rotation equivariant convolutional networks. *arXiv:1807.00583*, 2018.

[89] F. Jiang, A. Grigorev, S. Rho, Z. Tian, Y. Fu, W. Jifara, K. Adil, and S. Liu. Medical image semantic segmentation based on deep learning. *Neural Computing and Applications*, 29(5):1257–1265, 2018.

[90] G. Oliveira, C. Bollen, W. Burgard, and T. Brox. Efficient and robust deep networks for semantic segmentation. *International Journal of Robotics Research*, 37(4-5):472–491, 2018.

[91] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei. Fully convolutional adaptation networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[92] C. Liu, L. Chen, F. Schroff, H. Adam, W. Hua, A. Yuille, and F. Li. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. *arXiv preprint arXiv:1901.02985*, 2019.

[93] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[94] C. Yu, J. Wang, C. Peng, C. Gao, F. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[95] F. Lv, Q. Lian, G. Yang, G. Lin, S. J. Pan, and L. Duan. Domain adaptive semantic segmentation through structure enhancement. In *European Conference on Computer Vision (ECCV)*, 2018.

[96] J. Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv:1606.02585*, 2016.

[97] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. Semantic segmentation of aerial images with an ensemble of CNNs. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016.

[98] N. Audebert, B. L. Saux, and S. Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[99] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[100] N. Audebert, B. L. Saux, and S. Lefèvre. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140(June):20–32, 2018.

[101] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. High-resolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7092–7103, 2017.

[102] B. Kellenberger, M. Volpi, and D. Tuia. Learning class- and location-specific priors for urban semantic labeling with CNNs. In *Joint Urban Remote Sensing Event (JURSE)*, 2017.

[103] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135(January):158–172, 2018.

[104] M. Volpi and D. Tuia. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:48–60, 2018.

[105] M. Kampffmeyer, A. Salberg, and R. Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–9, 2016.

[106] N. Audebert, B. L. Saux, and S. Lefèvre. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sensing*, (4):page 1–18, 2017.

[107] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[108] M. Volpi and D. Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, 2017.

[109] D. Marcos, M. Volpi, B. Kellenberger, and D. Tuia. Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, DOI:10.1016/j.isprsjprs.2018.01.021.

[110] S. Srivastava, M. Volpi, and D. Tuia. Joint height estimation and semantic labeling of monocular aerial images with CNNs. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[111] L. Maggiolo, D. Marcos, G. Moser, and D. Tuia. Improving maps from CNNs trained with sparse, scribbled ground truths using fully connected CRFs. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.

[112] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck. Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:59–69, 2019.

[113] A. Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013.

[114] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[115] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[116] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, 2015.

[117] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[118] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[119] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, 2017.

[120] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[121] S. Lobry, J. Murray, D. Marcos, and D. Tuia. Visual question answering from remote sensing images. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019.

[122] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision (ECCV)*, 2018.

[123] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[124] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362, 2016.

[125] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[126] D. Marmanis, K. Schindler, J. D. W. amd S. Galliani, M. Datcu, and U. Stilla. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135:158–172, 2018.

[127] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *IEEE International Conference on Learning Representation (ICLR)*, 2015.

A Mou L., Ghamisi P., Zhu X., 2017. Deep Recurrent Neural Networks for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, 55(7), 3639-3655. Mou L., Ghamisi P., Zhu X., 2018. Corrections to “Deep Recurrent Neural Networks for Hyperspectral Image Classification”, IEEE Transactions on Geoscience and Remote Sensing, 56(2), 1214-1215.

Deep Recurrent Neural Networks for Hyperspectral Image Classification

Lichao Mou, *Student Member, IEEE*, Pedram Ghamisi, *Member, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*— In recent years, vector-based machine learning algorithms, such as random forests, support vector machines, and 1-D convolutional neural networks, have shown promising results in hyperspectral image classification. Such methodologies, nevertheless, can lead to information loss in representing hyperspectral pixels, which intrinsically have a sequence-based data structure. A recurrent neural network (RNN), an important branch of the deep learning family, is mainly designed to handle sequential data. Can sequence-based RNN be an effective method of hyperspectral image classification? In this paper, we propose a novel RNN model that can effectively analyze hyperspectral pixels as sequential data and then determine information categories via network reasoning. As far as we know, this is the first time that an RNN framework has been proposed for hyperspectral image classification. Specifically, our RNN makes use of a newly proposed activation function, parametric rectified tanh (PRetanh), for hyperspectral sequential data analysis instead of the popular tanh or rectified linear unit. The proposed activation function makes it possible to use fairly high learning rates without the risk of divergence during the training procedure. Moreover, a modified gated recurrent unit, which uses PRetanh for hidden representation, is adopted to construct the recurrent layer in our network to efficiently process hyperspectral data and reduce the total number of parameters. Experimental results on three airborne hyperspectral images suggest competitive performance in the proposed mode. In addition, the proposed network architecture opens a new window for future research, showcasing the huge potential of deep recurrent networks for hyperspectral data analysis.**

***Index Terms*— Convolutional neural network (CNN), deep learning, gated recurrent unit (GRU), hyperspectral image classification, long short-term memory (LSTM), recurrent neural network (RNN).**

I. INTRODUCTION

IN THE past few decades, the analysis of hyperspectral imagery acquired by remote sensors has attracted considerable attention in the remote sensing community, as such data are characterized in hundreds of continuous observation bands throughout the electromagnetic spectrum with high spectral resolution [1]. With this rich spectral information, different land cover categories can potentially be precisely differentiated. To benefit from this type of data, supervised hyperspectral image classification plays a significant role and has been investigated in many applications, including urban development [2]–[5], the monitoring of land changes [6]–[9], scene interpretation [10]–[13], and resource management [14], [15].

Numerous types of supervised classification models have been discussed in the literature, including decision trees [16], random forests [17], [18], and support vector machines (SVMs) [19], [20]. Among them, the random forest [18] develops multiple trees from randomly sampled subspaces of input hyperspectral pixel vectors and then combines the outputs via voting or a maximum *a posteriori* rule. In contrast, SVM, a supervised machine learning technique, has achieved great success in various applications and is considered a stable and efficient algorithm for hyperspectral image classification tasks. An SVM seeks to separate two-class data by learning an optimal decision hyperplane that can best separate the training samples in a kernel-included high dimensional feature space. Some strategies, such as one-against-all and one-against-one, enable the use of original binary SVM for multiclass classification. In addition, some extensions of the SVM model in hyperspectral image classification have been presented to improve the classification performance [21], [22].

When the ratio of the number of available training samples and the number of spectral bands is unbalanced, theoretical and practical problems may arise and the hyperspectral image classification becomes an ill-posed problem. For example, while keeping the number of available training samples constant, the classification accuracy will decrease when the dimension of input feature vectors becomes large [23], [24].

In recent years, deep learning has made promising achievements in the machine learning field [25]–[29]. It attempts to learn hierarchical representations from raw data and is capable of learning simple concepts first and then successfully building up more complex concepts by merging the simpler ones. In remote sensing, convolutional neural networks (CNNs) have been shown to be successful for hyperspectral data classification [30]–[32]. Hu *et al.* [30] presented a CNN that contains an input layer, a convolutional layer, a max-pooling layer, a fully connected layer, and an output layer for hyperspectral image classification. The CNN has been employed to classify hyperspectral data directly in the spectral domain. Makantasis *et al.* [31] presented a deep learning-based classification method that hierarchically constructs high-level features automatically. In particular, their model exploits

Manuscript received August 26, 2016; revised October 29, 2016; accepted November 15, 2016. Date of publication April 28, 2017; date of current version June 22, 2017. This work was supported in part by the China Scholarship Council, in part by the Alexander von Humboldt Foundation, in part by the Helmholtz Association through the framework of the Young Investigators Group SiPEO under Grant VH-NG-1018, and in part by the European Research Council (ERC) under the European Unions Horizon 2020 Research and Innovation Programme under Grant ERC-2016-StG-714087 (Acronym: So2Sat). *(Corresponding author: Xiao Xiang Zhu.)*

The authors are with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Wessling, Germany, and also with Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: lichao.mou@dlr.de; pedram.ghamisi@dlr.de; xiao.zhu@dlr.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2016.2636241

a CNN to encode the spectral and spatial information of pixels and a multilayer perceptron to conduct the classification task. Chen *et al.* [32] proposed a regularized 3-D CNN-based feature extraction model to extract efficient spectral-spatial features for hyperspectral image classification. In addition, Chen *et al.* [33] proposed a hybrid framework based on stacked autoencoders for the classification of the hyperspectral data.

All of the supervised models for hyperspectral images described earlier are vector-based methodologies.[1] It should be noted that these vector-based approaches can lead to information loss when representing hyperspectral pixels, which intrinsically have a sequence-based data structure. To the best of our knowledge, almost all advanced spectral classifiers, such as SVM, random forest, and CNN-based classification, are vector-based approaches, which consider hyperspectral data to be a collection of pixel vectors and perform the classification procedure in feature space: each pixel is considered a point in an orderless d-dimensional feature space in which d represents the number of dimensions (bands) [1]. However, hyperspectral data can be seen as a set of orderly and continuing spectra sequences in the spectral space. Analyzing hyperspectral imagery from a sequential perspective has not been addressed so far. Our motivation in this paper is to explore the representation of hyperspectral pixels via the sequential perspective instead of classifying hyperspectral data in the feature space.

In this paper, we make use of a recurrent neural network (RNN) to characterize the sequential property of a hyperspectral pixel vector for the classification task. An RNN [34]–[36] is a network that uses recurrent connections between neural activations at consecutive time steps; such a network uses hidden layers or memory cells to learn the states that model the underlying dynamics of the input sequence for sequential data over time. RNNs have gained significant attention for solving many challenging problems involving sequential data analysis, such as language modeling [37], machine translation [38], and speech recognition [39], [40]. Since the temporal variability of a sequential signal, such as a language sentence, is similar to the spectral variability of a hyperspectral pixel, the same idea can be applied to hyperspectral pixel vectors. The RNN exploits a recurrent procedure to characterize spectral correlation and band-to-band variability, where the network parameters are determined by training with available samples. In this context, we propose a novel RNN with a specially designed activation function and modified gated recurrent unit (GRU) to solve the multiclass classification for hyperspectral imagery. This paper contributes to the literature in three major respects.

1) We represent and process the pixels of hyperspectral images via a sequential perspective instead of taking them as feature vectors to capture the intrinsic sequence-based data structure of hyperspectral pixels. This enables us to take full advantage of the sequential property of hyperspectral data, e.g., spectral correlation and band-to-band variability.
2) An RNN with GRUs is proposed for a hyperspectral image classification task. To the best of our knowledge, this is the first use of the recurrent network model for the problem of hyperspectral image classification.
3) We introduce a new activation function, parametric rectified tanh (PRetanh), which generalizes the rectified unit for the deep RNN and then modifies the proposed activations of GRUs. With this new activation function, fairly high learning rates can be used to train the network without the risk of divergence.

The remainder of this paper is organized as follows. An introduction to RNNs is briefly given in Section II. The details of the proposed RNN architecture for hyperspectral image classification, including a novel activation function and modified GRU, are described in Section III. The network setup, experimental results, and a comparison with state-of-the-art approaches are provided in Section IV. Finally, Section V concludes this paper.

II. Background on Recurrent Neural Networks

An RNN [34], [35] is a class of artificial neural network that extends the conventional feedforward neural network with loops in connections. Unlike a feedforward neural network, an RNN is able to process the sequential inputs by having a recurrent hidden state whose activation at each step depends on that of the previous step. In this manner, the network can exhibit dynamic temporal behavior.

Given a sequence data $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_i$ is the data at ith time step, an RNN updates its recurrent hidden state $\mathbf{h}_t$ by

$$\mathbf{h}_t = \begin{cases} 0, & \text{if } t = 0 \\ \varphi(\mathbf{h}_{t-1}, \mathbf{x}_t), & \text{otherwise} \end{cases} \tag{1}$$

where φ is a nonlinear function, such as a logistic sigmoid function or hyperbolic tangent function. Optionally, the RNN may have an output $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$. For some tasks, such as hyperspectral image classification, we need only one output, i.e., $\mathbf{y}_T$.

In the traditional RNN model, the update rule of the recurrent hidden state in (1) is usually implemented as follows:

$$\mathbf{h}_t = \varphi(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1}) \tag{2}$$

where $\mathbf{W}$ and $\mathbf{U}$ are the coefficient matrices for the input at the present step and for the activation of recurrent hidden units at the previous step, respectively.

In fact, an RNN can model a probability distribution over the next element of the sequence data, given its present state $\mathbf{h}_t$, by capturing a distribution over sequence data of variable length. Let $p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ be the sequence probability, which can be decomposed into

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = p(\mathbf{x}_1) \cdots p(\mathbf{x}_T | \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}). \tag{3}$$

Then, each conditional probability distribution can be modeled with a recurrent network

$$p(\mathbf{x}_t | \mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) = \varphi(\mathbf{h}_t) \tag{4}$$

[1] Here, vector-based approaches refer to those that consider the input to be vectors. Although CNN-based models consider the inherent relationship of the inputs during the process, such models are still categorized as vector-based models in this paper.

where $\mathbf{h}_t$ is obtained from (1) and (4). Our motivation in this paper is apparent here: a hyperspectral pixel acts as sequential data instead of a feature vector, and so a recurrent network can be adopted to model the spectral sequence.

As an important branch of the deep learning family, RNNs have recently shown promising results in many machine learning and computer vision tasks. However, it has been observed that it is difficult to train the RNNs to deal with long-term sequential data, as the gradients tend to vanish. To address this issue, one common approach is to design a more sophisticated recurrent unit.

Long short-term memory (LSTM) [41], [42] is a special type of recurrent hidden unit, capable of learning long-term dependences. LSTM was initially introduced in [41]. Since then, a number of minor modifications to the original version have been made [42], [43]. A recurrent layer with traditional recurrent hidden units is shown in (2), which simply calculates a weighted linear sum of inputs and then applies a nonlinear function. In contrast, an LSTM-based recurrent layer creates a memory cell $\mathbf{c}_t$ at step t. The activation of the LSTM units can be computed by

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t) \tag{5}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function and $\mathbf{o}_t$ is the output gate that determines the part of the memory content that will be exposed. The output gate is updated by

$$\mathbf{o}_t = \sigma(\mathbf{W}_{oi}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{W}_{oc}\mathbf{c}_t) \tag{6}$$

where $\sigma(\cdot)$ is a logistic sigmoid function and $\mathbf{W}$ terms denote weight matrices; e.g., $\mathbf{W}_{oi}$ is the input–output weight matrix and $\mathbf{W}_{oc}$ represents the memory-output weight matrix.

The memory cell $\mathbf{c}_t$ is updated by adding new content of memory cell $\tilde{\mathbf{c}}_t$ and discarding part of the present memory content

$$\mathbf{c}_t = \mathbf{i}_t \odot \tilde{\mathbf{c}}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} \tag{7}$$

where $\odot$ is an elementwise multiplication, and the new content of memory cell $\tilde{\mathbf{c}}_t$ is obtained by

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_{ci}\mathbf{x}_t + \mathbf{W}_{ch}\mathbf{h}_{t-1}). \tag{8}$$

Input gate $\mathbf{i}_t$ modulates the extent to which the new memory information is added to the memory cell. The degree to which content of the existing memory cell is forgotten is decided by the forget gate $\mathbf{f}_t$. The equations that calculate these two gates are as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{W}_{ic}\mathbf{c}_{t-1}) \tag{9}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fi}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{W}_{fc}\mathbf{c}_{t-1}). \tag{10}$$

Fig. 1 shows the graph model of LSTM.

III. Proposed Recurrent Network for Hyperspectral Image Classification

In the main procedure of the proposed recurrent network, as shown in Fig. 2, the input of the network is a hyperspectral pixel $\mathbf{x}$, where the kth spectral band is denoted as x^k. The output is a label that indicates the category the pixel belongs to. The entire classification map can be obtained by applying the network to all pixels in the image. The flowchart of our RNN can be summarized as follows.

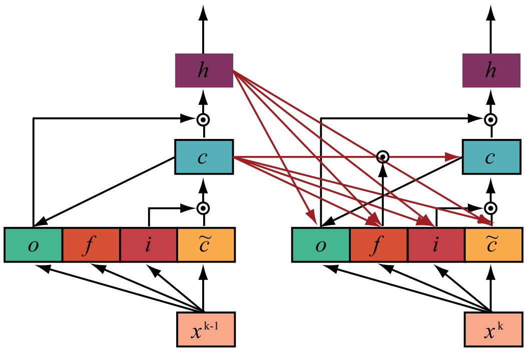


Fig. 1. Graphic model of LSTM. i, f, o, and c are the input gate, forget gate, output gate, and memory cell, respectively. The new memory cell content is denoted by $\tilde{c}$.

1) First, the value of the existing spectral band x^k is fed into the input layer.
2) Then, the recurrent layer receives x^k and calculates the hidden state information for the current band; it also restores that information in the meantime.
3) Subsequently, the value of the next band x^{k+1} is input to the recurrent layer simultaneously with the state information of x^k, and the activation at spectral band $k+1$ is computed by a linear interpolation between proposal activation and the activation of the previous band k.
4) Finally, the RNN predicts the label of the input hyperspectral pixel by looping through the entire hyperspectral pixel sequence.

Two important factors affect the performance of RNN: the activation function and the structure of the recurrent unit. In Section III-A and Section III-B, we will discuss our innovative contributions on these two factors in detail.

A. Parametric Rectified tanh

Recently, rectified linear activation functions, such as the rectified linear unit (ReLU) [25], have become a common approach to training deep convolutional networks. They have been proposed to alleviate the vanishing gradient problem and speed up the learning process by identifying positive values; however, this leads to a nonbounded output. We have utilized the proposed activation function instead of the existing ones for several reasons.

1) To train an RNN in our task, the vanishing gradient problem is not a concern, as modern recurrent network models, such as LSTM and GRU, have already been designed to tackle this issue. By using gates, LSTMs and the GRUs help preserve the errors that can be backpropagated through sequence and layers. By maintaining a more constant error, they allow recurrent networks to continue to learn over many bands of hyperspectral pixels without the risk of the vanishing gradient.
2) In our experiments, the recurrent network often runs into numerical problems when a rectified linear function like

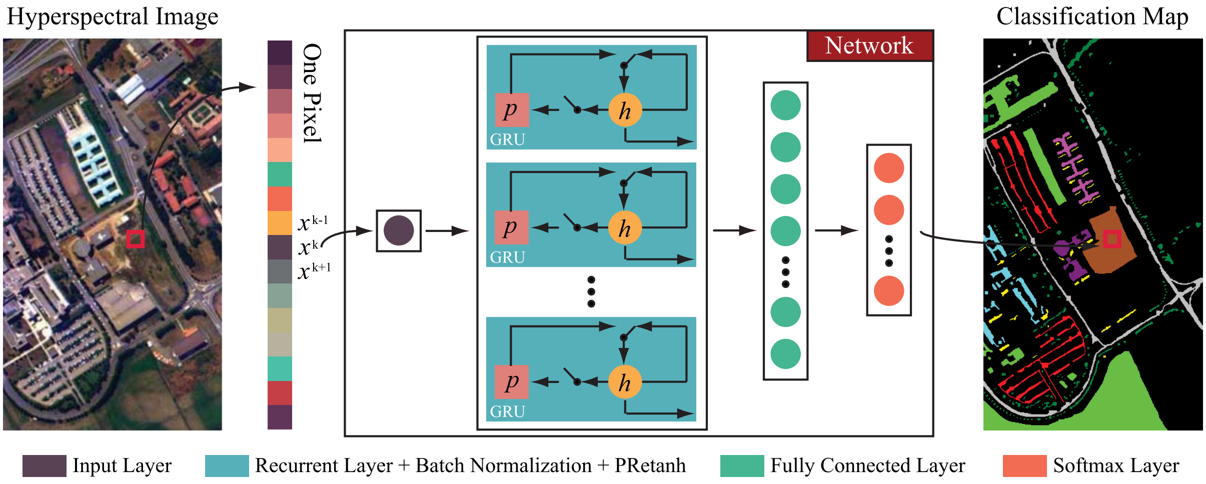


Fig. 2. Overview of our pipeline. First, the value of existing spectral band x^k is fed into the input layer. Then, the recurrent layer receives x^k and calculates the hidden state information for the current band; it also restores that information in the meantime. Next, the value of the next band x^{k+1} is input to the recurrent layer simultaneously with the state information of x^k, and the activation at spectral band $k+1$ is computed by a linear interpolation between proposal activation and the activation of previous band k. Finally, the RNN can predict the label of the input hyperspectral pixel by looping through the entire hyperspectral pixel sequence.

ReLU is used as an activation functions for the network output, given that gradients often need to be truncated often (and ReLU cannot dampen them like the bounded activation functions, such as tanh).

3) Traditional bounded activation functions, such as sigmoid and tanh, are always likely to generate some nonzero values, resulting in dense representations, while sparse representations seem to be better than dense representations in terms of representation learning.

Thus, to train a valid recurrent network for the hyperspectral image classification, we designed the new activation function PRetanh, which has two major advantages: 1) producing a bounded output and 2) promoting sparsity adaptively.

Definition: In this section, we introduce a newly defined activation function—PRetanh. It is defined as

$$f(h_i) = \begin{cases} \tanh(h_i), & \text{if } h_i > 0 \\ \lambda_i \tanh(h_i), & \text{if } h_i \leq 0 \end{cases} \tag{11}$$

where h_i is the input of the nonlinear activation f on the ith channel and $0 \leq \lambda_i \leq 1$ is a coefficient that can control the range of the negative part. The subscript i means that PRetanh can be varied in different channels. When $\lambda_i = 0$, it turns to

$$f(h_i) = \max(0, f(h_i)) = \max(0, \tanh(h_i)). \tag{12}$$

When λ_i is a learnable parameter, we refer to (11) as a parametric rectified hyperbolic tangent function. Fig. 3 shows the shapes of tanh and PRetanh. Equation (11) is equivalent to

$$f(h_i) = \max(0, \tanh(h_i)) + \lambda_i \min(0, \tanh(h_i)). \tag{13}$$

In our method, the PRetanh parameter λ_i is adaptively learned jointly with the whole neural network model. We expect that end-to-end training can lead to more specialized activations. Note that extra parameters are introduced in PRetanh. The total number of extra parameters for each layer is equal to the number of channels, which is negligible when taking into account the number of weights of the whole network. Therefore, we anticipate no extra risk of overfitting with the same number of training samples. In addition, a channel-shared variant version of PRetanh can be considered

$$f(h_i) = \max(0, \tanh(h_i)) + \lambda \min(0, \tanh(h_i)) \tag{14}$$

where all channels of one layer share the same coefficient λ. In this case, only a single extra parameter is introduced for each layer.

Optimization: With respect to the training of PRetanh, we use the backpropagation algorithm and simultaneously optimize the parameters of PRetanh with the neural networks. Suppose we have an objective function L that we wish to minimize, and the update rule of parameter λ_i is derived by the chain rule

$$\frac{\partial L}{\partial \lambda_i} = \sum_{h_i} \frac{\partial L}{\partial f(h_i)} \frac{\partial f(h_i)}{\partial \lambda_i}. \tag{15}$$

The term $(\partial L)/(\partial f(h_i))$ is the gradient backpropagated from the deeper layer of PRetanh. The summation $\sum_{h_i}$ is applied in all positions of the feature maps. Specifically, the gradient of activation is given by

$$\frac{\partial f(h_i)}{\partial \lambda_i} = \begin{cases} 0, & \text{if } h_i > 0 \\ \tanh(h_i), & \text{if } h_i \leq 0. \end{cases} \tag{16}$$

Equation (16) can be rewritten as follows:

$$\frac{\partial f(h_i)}{\partial \lambda_i} = \min(0, \tanh(h_i)). \tag{17}$$

Moreover, for the channel-shared variant version, the gradient of λ is as follows:

$$\frac{\partial L}{\partial \lambda} = \sum_i \sum_{h_i} \frac{\partial L}{\partial f(h_i)} \frac{\partial f(h_i)}{\partial \lambda} \tag{18}$$

where $\sum_i$ sums over all channels of the layer.

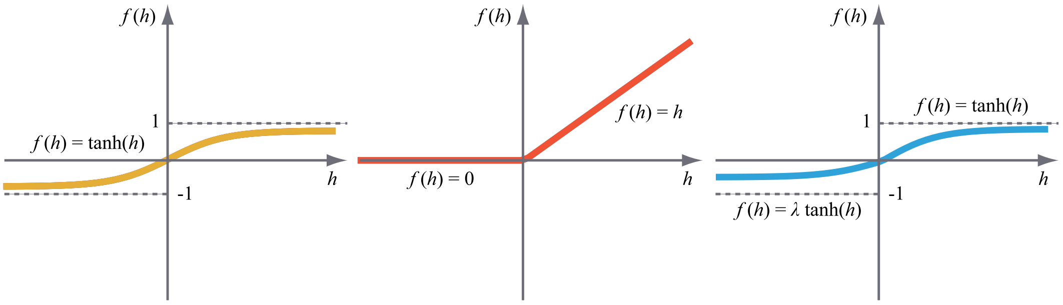


Fig. 3. (From left to right) tanh, ReLU, and PRetanh. For PRetanh, the coefficient λ of the negative part is not constant and is adaptively learned.

The momentum method is commonly used to help accelerate stochastic gradient descent in the relevant direction and dampens oscillations by adding a fraction γ of the update parameter. Here, we adopt the momentum method when updating parameter λ_i. The updating rule is

$$\Delta\lambda_i := \gamma\,\Delta\lambda_i + \eta\frac{\partial L}{\partial\lambda_i} \tag{19}$$

where η is the learning rate and γ is the momentum. Note that we do not use weight decay, i.e., ℓ_2 regularization, for updating λ_i, since a weight decay term tends to push the rectified parameters λ_i to zero.

Analysis: The two major advantages lie in obtaining adaptively sparse and bounded output. Sparsity arises when $\lambda_i = 0$ and $h_i \leq 0$. The more such units exist in a layer, the more sparse the resulting representations will be. The traditional tanh, in contrast, is always likely to generate some nonzero values, resulting in dense representations, while sparse representations seem to be better than dense representations. In addition, unlike the popular ReLU [25], which restricts the form of the negative part, we do not apply any constraints or regularization to it. As a result, the parameter λ_i that controls sparsity can be learned freely as the network trains. The other merit of PRetanh, the bounded output, is important from a practical perspective, because it means that the activations of the recurrent network will not blow up. The bounded output can reduce the probability of change in the distribution of internal nodes of deep networks to some extent, which allows fairly high learning rates to be used without the risk of divergence. In Section IV, it will be demonstrated that, compared with ReLU, using PRetanh as the activation function can effectively overcome the divergence of the recurrent network for hyperspectral image classification in the course of training.

Nevertheless, since PRetanh is affected by tanh, it likely moves many inputs into the saturated regime of the nonlinearity, and slows down the convergence. This effect is amplified as the recurrent network depth increases. In practice, the saturation problem and the resulting vanishing gradients are usually addressed by a carefully chosen initialization and the use of small learning rates. If, however, the distribution of inputs could be ensured to remain more stable as the training goes on, the optimizer would be less likely to fall into the saturation regime. In this paper, we combine a batch normalization technique with PRetanh to avoid the vanishing gradient problem.

For a layer with D dimensional output $\mathbf{h} = (h_1, h_2, \ldots, h_D)$, we normalize the ith channel as follows:

$$\hat{h}_i = \frac{h_i - \mathrm{E}[h_i]}{\sqrt{\mathrm{Var}[h_i]}} \tag{20}$$

where $\mathrm{E}[h_i] = \frac{1}{N}\sum_{j=1}^{N} h_i^{(j)}$ is the expectation, $\mathcal{H} = \{h_i^{(1)\ldots(N)}\}$ represents the set of values of h_i over a mini training batch, and $\mathrm{Var}[h_i]$ is the variance. We also need to scale and shift the normalized values; otherwise, just normalizing a layer would limit the layer in terms of what it can represent. Therefore, the normalized input h_i is transformed into

$$g(h_i) = \alpha_i \hat{h}_i + \beta_i \tag{21}$$

where α and β are parameters learned along with the original network parameters. Finally, batch normalization makes it possible to use PRetanh nonlinearities by preventing the network from getting stuck in the saturated modes.

B. Recurrent Unit

Recently, more and more empirical results have demonstrated that RNNs are not just powerful in theory [42], [44], [45] but can also be reliably learned in practice for processing long-term sequential data [36], [46], [47]. One interesting observation is that a few of these successes were obtained with the traditional RNN model. Rather, they used an RNN with sophisticated recurrent hidden units like LSTM, because such structures are capable of alleviating the vanishing gradient problem. However, available training samples for remote sensing image classification are often limited, forcing researchers to control the total number of trainable parameters of the network as much as possible. We, therefore, design a novel GRU with PReLU, which is able to deal with long-term sequential data like hyperspectral sequences and is more suitable for a small number of training samples, since it has fewer parameters than LSTM.

1) LSTM for Hyperspectral Image Classification: For a hyperspectral image classification task, given a hyperspectral pixel sequence $\mathbf{x} = (x^1, x^2, \ldots, x^K)$, a traditional RNN framework [34] calculates the hidden vector sequence $\mathbf{h} = (\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^K)$ by iterating the following equation from $k = 1$ to K:

$$\mathbf{h}^k = \varphi(\mathbf{w}_{ih}x^k + \mathbf{W}_{hh}\mathbf{h}^{k-1} + \mathbf{b}_h) \tag{22}$$

where $\mathbf{w}_{ih}$ denotes the input-hidden weight vector, $\mathbf{W}_{hh}$ represents the context weight matrix of the hidden layer, $\mathbf{b}_h$ is the hidden bias vector, and $\varphi(\cdot)$ is the hidden layer activation function. Finally, the predicted label $\mathbf{y}$ can be computed as follows:

$$\mathbf{y} = \mathbf{W}_{oh}\mathbf{h}^K + \mathbf{b}_o \tag{23}$$

where $\mathbf{W}_{oh}$ is the output-hidden weight matrix and $\mathbf{b}_o$ is the bias vector of output layer.

In this paper, we want to use an RNN to characterize the spectral correlation and band-to-band variability when mapping between input pixel sequences and output labels. Unfortunately, for standard recurrent network architecture, the range of spectral contexts that can be accessed in practice is quite limited. The problem is that the influence of a given input on the hidden layer and, therefore, on the network output either decays or blows up exponentially as it cycles around the recurrent connections of the network. This effect is a common challenge in designing and training deep RNNs and is known as the vanishing gradient problem [48].

To process long-term sequences, which is crucial to the task, as hyperspectral imagery usually includes hundreds of spectral bands, LSTMs were proposed to address the vanishing gradient problem. LSTMs [42] introduce the gate concept and memory cell to help preserve the error that can be backpropagated through steps and layers. By maintaining a more constant error, they allow recurrent networks to continue to learn over many steps (over 1000) and thereby enable us to utilize a large range of spectral contexts, e.g., to link the first and last spectral bands remotely.

2) Gated Recurrent Unit With PReLU: However, LSTMs lead to more parameters, which need to be learned. And, as discussed earlier, the limited number of training samples drives a need to restrict the number of parameters, to avoid overfitting.

Therefore, a deep RNN with modified GRUs tailored for hyperspectral sequence analysis is proposed for hyperspectral image classification. GRUs [44], [45] have fewer parameters than LSTMs, and can also effectively process a long-term spectral sequence. Moreover, PReLU is introduced to our modified GRUs, allowing us to use fairly high learning rates without the risk of divergence.

A GRU can cause a recurrent unit to adaptively capture the dependences of different spectral bands. Similar to the LSTM unit, the GRU has gate units that control the flow of information inside the unit without including separate memory cells.

The activation h_i^k of the ith GRU at spectral band k is computed by a linear interpolation between the proposal activation p_i^k and the activation of the previous spectral band h_i^{k-1}

$$h_i^k = u_i^k p_i^k + \left(1 - u_i^k\right) h_i^{k-1} \tag{24}$$

where u_i^k is an update gate that determines how much the unit updates its activation or content. The update gate u_i^k can be calculated as follows:

$$u_i^k = \sigma(\mathbf{w}_{ui}x^k + \mathbf{W}_{uh}\mathbf{h}^{k-1})_i \tag{25}$$

where $\mathbf{w}_{ui}$ is the input-update weight vector and $\mathbf{W}_{uh}$ represents the update-hidden weight matrix.

Similarly to LSTM, the GRU takes a linear sum between the newly computed state and the present state. However, it lacks a mechanism to control what part of the state information will be exposed, rather exposing the whole state value at each spectral band.

The proposal activation p_i^k is computed using the value of the existing spectral band and the activation of the previous band, which reflects the updated information of the recurrent hidden state. It is calculated with PRetanh and batch normalization as follows:

$$p_i^k = f(g(\mathbf{w}_{pi}x^k + \mathbf{W}_{rh}(\mathbf{r}^k \odot \mathbf{h}^{k-1})))_i \tag{26}$$

where $\mathbf{r}^k$ is a set of reset gates, $\mathbf{w}_{pi}$ denotes the proposal-input weight vector, and $\mathbf{W}_{rh}$ represents the reset-hidden weight matrix. Moreover, $f(\cdot)$ and $g(\cdot)$ represent PRetanh and batch normalization, respectively. When the reset gate r_i^k is fully OFF, i.e., r_i^k is 0, it will completely discard the activation of the hidden layer at previous spectral bands h_i^{k-1} and only use the value of the existing spectral band x^k. When open (r_i^k close to 1), in contrast, the reset gate will partially keep the information of the previous step.

Let $\hat{p}_i^k = \mathbf{w}_{pi}x^k + \mathbf{W}_{rh}(\mathbf{r}^k \odot \mathbf{h}^{k-1})$. Equation (26) can then be transformed as

$$\begin{aligned} p_i^k = \max &\left(0, \tanh\left(\alpha_i \frac{\hat{p}_i^k - \mathrm{E}\left[\hat{p}_i^k\right]}{\sqrt{\mathrm{Var}\left[\hat{p}_i^k\right]}} + \beta_i\right)\right) \\ &+ \lambda_i \min\left(0, \tanh\left(\alpha_i \frac{\hat{p}_i^k - \mathrm{E}\left[\hat{p}_i^k\right]}{\sqrt{\mathrm{Var}\left[\hat{p}_i^k\right]}} + \beta_i\right)\right). \end{aligned} \tag{27}$$

The reset gate r_i^k is computed similar to the update gate

$$r_i^k = \sigma(\mathbf{w}_{ri}x^k + \mathbf{W}_{rh}\mathbf{h}^{k-1})_i \tag{28}$$

where $\mathbf{w}_{ri}$ and $\mathbf{W}_{rh}$ are the reset-input weight vector and the reset-hidden weight matrix, respectively.

Fig. 4 shows the graphic model of the GRU through time.

IV. Experimental Results and Discussion

A. Data Description

1) Pavia University: This data set is acquired by reflective optics system imaging spectrometer (ROSIS). The image is of 610 × 340 pixels covering the Engineering School at the University of Pavia, which was collected under the HySens project managed by the German Aerospace Agency (DLR). The ROSIS-03 sensor comprises 115 spectral channels ranging from 430 to 860 nm. In this data set, 12 noisy channels have been removed and the remaining 103 spectral channels are investigated in this paper. The spatial resolution is 1.3 m per pixel. The available training samples of this data set cover nine classes of interests. Table I provides information about different classes and their corresponding training and test samples.

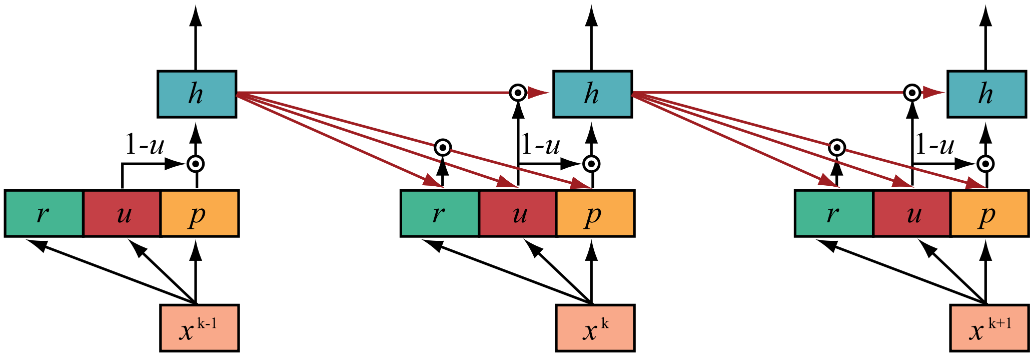


Fig. 4. Graphic model of a GRU through time. The reset and update gates are denoted by r and u, respectively, and p and h are the proposal activation and the final activation.

TABLE I
NUMBER OF TRAINING AND TEST SAMPLES USED IN THE PAVIA UNIVERSITY DATA SET

Class No.	Class Name	Training	Test
1	Asphalt	548	6631
2	Meadows	540	18649
3	Gravel	392	2099
4	Trees	524	3064
5	Metal sheets	265	1345
6	Bare Soil	532	5029
7	Bitumen	375	1330
8	Bricks	514	3682
9	Shadows	231	947
	TOTAL	3921	42776

TABLE II
NUMBER OF TRAINING AND TEST SAMPLES USED IN THE HOUSTON DATA SET

Class No.	Class Name	Training	Test
1	Grass Healthy	198	1053
2	Grass Stressed	190	1064
3	Grass Synthetic	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot 1	192	1041
13	Parking Lot 2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
	TOTAL	2832	12197

TABLE III
NUMBER OF TRAINING AND TEST SAMPLES USED IN THE INDIAN PINES DATA SET

Class No.	Class Name	Training	Test
1	Alfalfa	50	1384
2	Corn-notill	50	784
3	Corn-min	50	184
4	Corn	50	447
5	Grass-pasture	50	697
6	Grass-trees	50	439
7	Grass-pasture-mowed	50	918
8	Hay-windrowed	50	2418
9	Oats	50	564
10	Soybean-notill	50	162
11	Soybean-mintill	50	1244
12	Soybean-clean	50	330
13	Wheat	50	45
14	Woods	15	39
15	Buildings-grass-trees	15	11
16	Stone-steel-towers	15	5
	TOTAL	695	9671

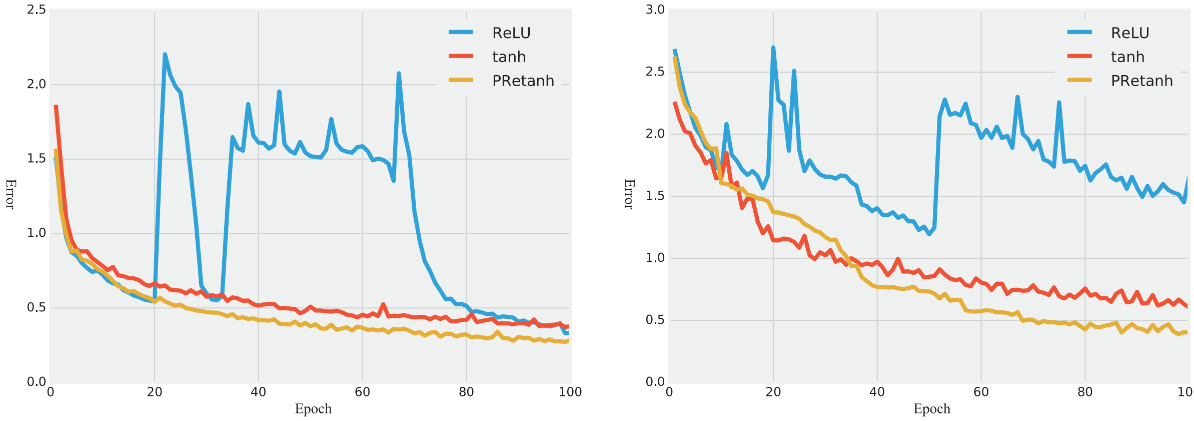
2) Houston Data: The second data set was acquired over the University of Houston campus and its neighboring urban area. It was collected with an ITRES-CASI 1500 sensor on June 23, 2012 between 17:37:10 and 17:39:50 UTC. The average altitude of the sensor was about 1676 m, which results in 2.5-m spatial resolution data consisting of 349 by 1905 pixels. The hyperspectral imagery consists of 144 spectral bands ranging from 380 to 1050 nm and was processed (radiometric correction, attitude processing, GPS processing, geocorrection, and so on) to yield the final geocorrected image cube representing the sensor spectral radiance. Table II provides information about all 15 classes of this data set with their corresponding training and test samples.

3) Indian Pines Data: The third data set was gathered by an airborne visible/infrared imaging spectrometer sensor over the Indian Pines agricultural site in Northwestern Indiana in June 1992, and presents 16 classes, mostly related to land covers. The data set consists of 145 by 145 pixels with a spatial resolution of 20 m per pixel and 10-nm spectral resolution over the range of 400–2500 nm. In this paper, we made use of 200 bands, after removing 20 bands affected by atmosphere absorption. The number of training and test samples is displayed in Table III.

B. General Information

To evaluate the performance of different models for hyperspectral image classification, we use the following evaluation criteria.

1) *Overall Accuracy (OA):* This index shows the number of hyperspectral pixels that are classified correctly, divided by the number of test samples.
2) *Average Accuracy (AA):* This measure is the average value of the classification accuracies of all classes.



(a)

(b)

Fig. 5. Learning curves for a recurrent network with ReLU, one with tanh and one with the proposed PRetanh on the training samples of (a) Pavia University data set and (b) Houston data set. As shown in these figures, with PRetanh, we can make use of a fairly high learning rate, e.g., 1.0 instead of a relatively low default 0.002, to train the recurrent network for hyperspectral image classification without the risk of divergence. Meanwhile, it can be seen that the ReLU can cause the recurrent network to diverge when a fairly high learning rate is used. Here, we use the Adadelta optimization algorithm.

TABLE IV

Classification Accuracies of Different Techniques in Percentages for Pavia University. The Best Accuracy in Each Row Is Shown in Bold

Class No.	Class Name	RF-200	SVM-RBF	CNN	RNN-LSTM	RNN-GRU-tanh	RNN-GRU-PRetanh
1	Asphalt	80.85	80.80	83.73	77.45	78.42	**84.45**
2	Meadows	55.29	66.78	65.70	61.83	69.17	**85.24**
3	Gravel	52.93	**73.18**	67.03	64.60	47.83	54.31
4	Trees	**98.79**	95.17	94.03	97.98	97.16	95.17
5	Metal Sheets	99.26	99.55	99.41	99.18	97.84	**99.93**
6	Bare Soil	78.76	92.90	**96.30**	91.19	85.86	80.99
7	Bitumen	84.36	90.08	**93.83**	90.90	86.84	88.35
8	Bricks	91.58	91.20	93.56	92.29	**94.27**	88.62
9	Shadows	98.20	93.77	99.79	97.47	94.93	**99.89**
OA	-	71.37	78.82	80.51	77.99	80.70	**88.85**
AA	-	82.23	87.05	**88.15**	85.88	83.59	86.33
Kappa	-	0.6484	0.7358	0.7423	0.7028	0.7201	**0.8048**

3) *Kappa Coefficient:* This metric is a statistical measurement of agreement between the final classification map and the ground-truth map. It is the percentage agreement corrected by the level of agreement that could be expected due to chance alone. It is generally thought to be a more robust measure than a simple percent agreement calculation, since k takes into account the agreement occurring by chance [1].

If the number of samples for each category is identical, OA and AA are equal. However, the category distribution suffers from an imbalanced phenomenon in practice. Adopting OA alone is not precise, since rare categories are commonly ignored. Therefore, AA is also utilized to evaluate the performance of different classification models. Strong differences between the OA and AA may indicate that a specific class is incorrectly classified with a high proportion.

To validate the effectiveness of the proposed RNN-based classification framework, it is compared with the most widely used vector-based classification models, SVM and random forest. The SVM with an RBF kernel was implemented using the libsvm package.[2] Fivefold cross-validation is taken into account to tune the hyperplane parameters. Furthermore, in this paper, experiments using other popular activation functions (i.e., tanh and ReLU) and recurrent units (i.e., LSTM) are also carried out to verify the validity of the proposed network. To conduct a fair comparison, PRetanh/tanh/ReLU models are trained using the same total number of epochs, and the same network architecture is adopted. The learning rates are also switched after running the same number of epochs. The methods included in the comparison are summarized as follows.

1) *RF-200:* Random forest with 200 trees.
2) *SVM-RBF:* RBF kernel SVM with cross validation.
3) *CNN:* The architecture of the CNN is set as in [30], and contains an input layer, a convolution layer, a max-pooling layer, a fully connected layer, and an output layer. The number of convolutional kernels is 20 for all three data sets. The length of each convolution kernel and pooling size is 11 and 3, respectively. Furthermore, 100 hidden units are included in the fully connected layer.
4) *RNN-LSTM:* RNN with LSTM recurrent units. We follow the implementation of LSTM as used in [42].

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

TABLE V

CLASSIFICATION ACCURACIES OF TEST SAMPLES ON THE HOUSTON DATA SET. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	Class Name	RF-200	SVM-RBF	CNN	RNN-LSTM	RNN-GRU-tahn	RNN-GRU-PRetanh
1	Grass Healthy	**82.62**	81.96	81.20	81.86	82.43	82.53
2	Grass Stressed	83.36	80.55	**83.55**	81.20	82.42	83.36
3	Grass Synthetic	98.02	99.80	99.41	99.41	97.23	**100**
4	Tree	91.76	**92.23**	91.57	90.06	89.30	90.53
5	Soil	97.06	97.63	94.79	93.09	78.22	**97.82**
6	Water	**99.30**	95.10	95.10	96.50	95.10	93.01
7	Residential	75.37	**76.59**	63.53	73.41	70.43	75.37
8	Commercial	32.95	35.52	**42.64**	34.09	32.57	42.36
9	Road	67.14	70.44	58.17	62.61	70.25	**77.62**
10	Highway	43.73	**60.04**	41.80	39.96	43.24	57.63
11	Railway	70.11	76.57	75.71	60.44	69.07	**77.42**
12	Parking Lot 1	54.95	73.10	**84.15**	65.42	50.72	69.74
13	Parking Lot 2	59.65	**68.77**	40.00	58.95	58.25	66.32
14	Tennis Court	99.19	**100**	98.79	96.76	97.98	**100**
15	Running Track	97.67	**98.10**	97.89	88.37	96.83	95.98
OA	-	72.93	77.09	85.42	85.41	85.73	**89.85**
AA	-	76.86	80.43	76.55	74.81	74.27	**80.65**
Kappa	-	0.7091	0.7536	0.7200	0.6889	0.6785	**0.7606**

TABLE VI

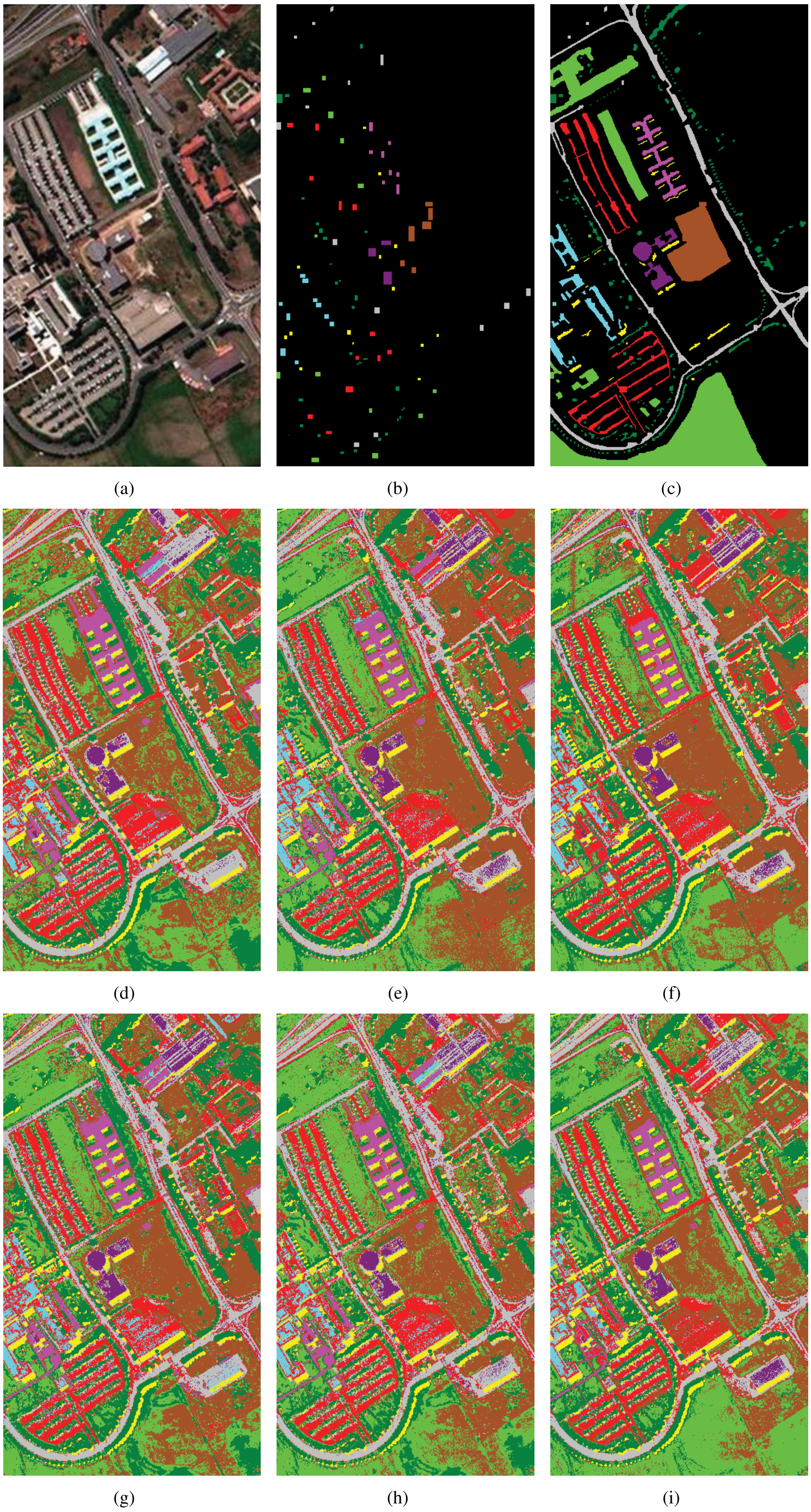
ACCURACY COMPARISON FOR THE INDIAN PINES DATA SET. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	Class Name	RF-200	SVM-RBF	CNN	RNN-LSTM	RNN-GRU-tahn	RNN-GRU-PRetanh
1	Alfalfa	54.84	60.77	56.79	46.03	68.93	**70.59**
2	Corn-notill	58.42	**77.68**	52.17	61.73	40.94	70.28
3	Corn-min	82.61	79.35	85.33	**86.96**	78.80	81.52
4	Corn	85.91	**91.05**	87.92	87.02	87.92	90.16
5	Grass-pasture	80.49	84.36	85.22	86.66	87.52	**91.97**
6	Grass-trees	94.76	92.03	**97.49**	**97.49**	97.27	96.13
7	Grass-pasture-mowed	77.34	69.61	74.62	59.69	82.79	**84.75**
8	Hay-windrowed	59.43	59.31	**67.99**	64.89	50.58	59.64
9	Oats	63.48	79.61	58.87	60.46	79.43	**86.17**
10	Soybean-notill	95.06	97.53	98.77	98.77	98.77	**99.38**
11	Soybean-mintill	**88.26**	85.21	87.62	75.32	84.73	84.97
12	Soybean-clean	54.85	63.64	72.42	71.82	61.21	**77.58**
13	Wheat	97.78	**100**	93.33	91.11	88.89	95.56
14	Woods	58.97	**87.18**	71.79	79.49	79.49	84.62
15	Buildings-grass-trees	81.82	**90.91**	**90.91**	**90.91**	**90.91**	**90.91**
16	Stone-steel-towers	**100**	**100**	**100**	**100**	**100**	**100**
OA	-	69.79	72.78	84.18	80.52	85.71	**88.63**
AA	-	77.13	82.39	80.08	78.65	79.89	**85.26**
Kappa	-	0.6589	0.6931	0.6852	0.6372	0.6633	**0.7366**

5) *RNN-GRU-tanh:* RNN with GRUs that use tanh as the activation function.
6) *RNN-GRU-ReLU:* ReLU is adopted to activate the output of recurrent units.
7) *RNN-GRU-PRetanh:* Our final network uses the proposed PRetanh activation function for the hidden representation of GRUs.

To make the proposed approach fully comparable with other supervised classifiers, we used the standard sets of training and test samples for the data sets. For instance, we used the training and test sets of the 2013 GRSS Fusion Contest for the classification of the Houston data.

The RNN was trained with the Adadelta algorithm, and all the suggested default parameters except the learning rate were used for all of the following experiments. We made use of a fairly high learning rate of 1.0 instead of the relatively low default of 0.002 to train the network. The proposed network model uses a single recurrent layer that adopts our modified GRUs of size 64 with sigmoid gate activation and PRetanh activation functions for hidden representations. The output layer uses softmax activation and then outputs a one-hot vector for hyperspectral image classification. All weight matrices in our RNN and bias vectors are initialized with a uniform distribution, and the values of these weight matrices and bias vectors are initialized in the range $[-0.1,0.1]$. Then, all the weights can be updated during the training procedure. In both hyperspectral data sets, we randomly chose 10% of the training samples as the validation set. That is, during the training, we used 90% of the training samples to learn the parameters, including the weight matrices $\mathbf{W}$, bias vectors $\mathbf{b}$, the parameters α and β of batch normalization, and the coefficients λ of PRetanh, and used the remaining 10% of the training samples as validation to tune the superparameters, such as the number of recurrent units in the recurrent layer. All test samples were used to evaluate the final performance of the trained recurrent



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6. Classification results obtained by different methods for the Pavia University scene. (a) Composite image of hyperspectral data. (b) Training data. (c) Ground-truth reference. (d) RF-200. (e) SVM-RBF. (f) CNN. (g) RNN-LSTM. (h) RNN-GRU-tanh. (i) RNN-GRU-PRetanh.

network. It is noteworthy that the Indian Pines data are a small and unbalanced data set, which is challenging for training a valid supervised recurrent network. To address this concern, we not only use a dropout with a probability of 0.5 on the output of recurrent layer but also utilize a dropout of 0.2 on the weight matrices of the network, which indicates the

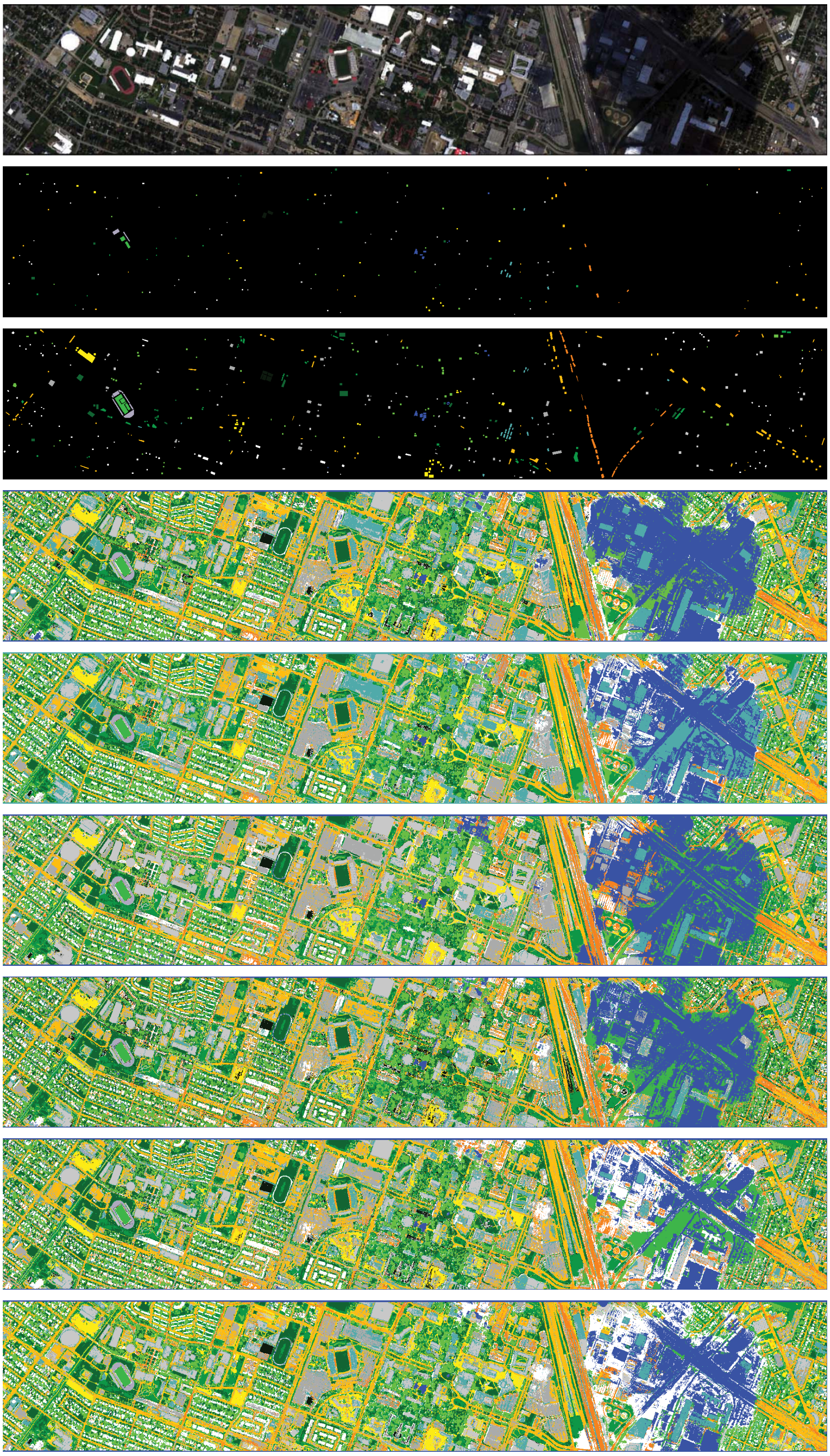


Fig. 7. Classification results obtained by different methods for the Houston scene. (From top to bottom) True-color composite of the hyperspectral data (wavelength R: 640.7 nm, G: 550.2 nm, and B: 459.6 nm), training data, ground-truth reference, RF-200, SVM-RBF, CNN, RNN-LSTM, RNN-GRU-tanh, and RNN-GRU-PRetanh.

fraction of the input units to drop for input gates and recurrent connections.

The experiments are organized into three parts. The first one aims primarily at analyzing the behavior of different activation functions, which involves tanh, ReLU, and the proposed PRetanh. The comparison of the LSTM unit and GRU is also discussed in this part. In the second experiment, the effectiveness of an RNN that is based on the sequential

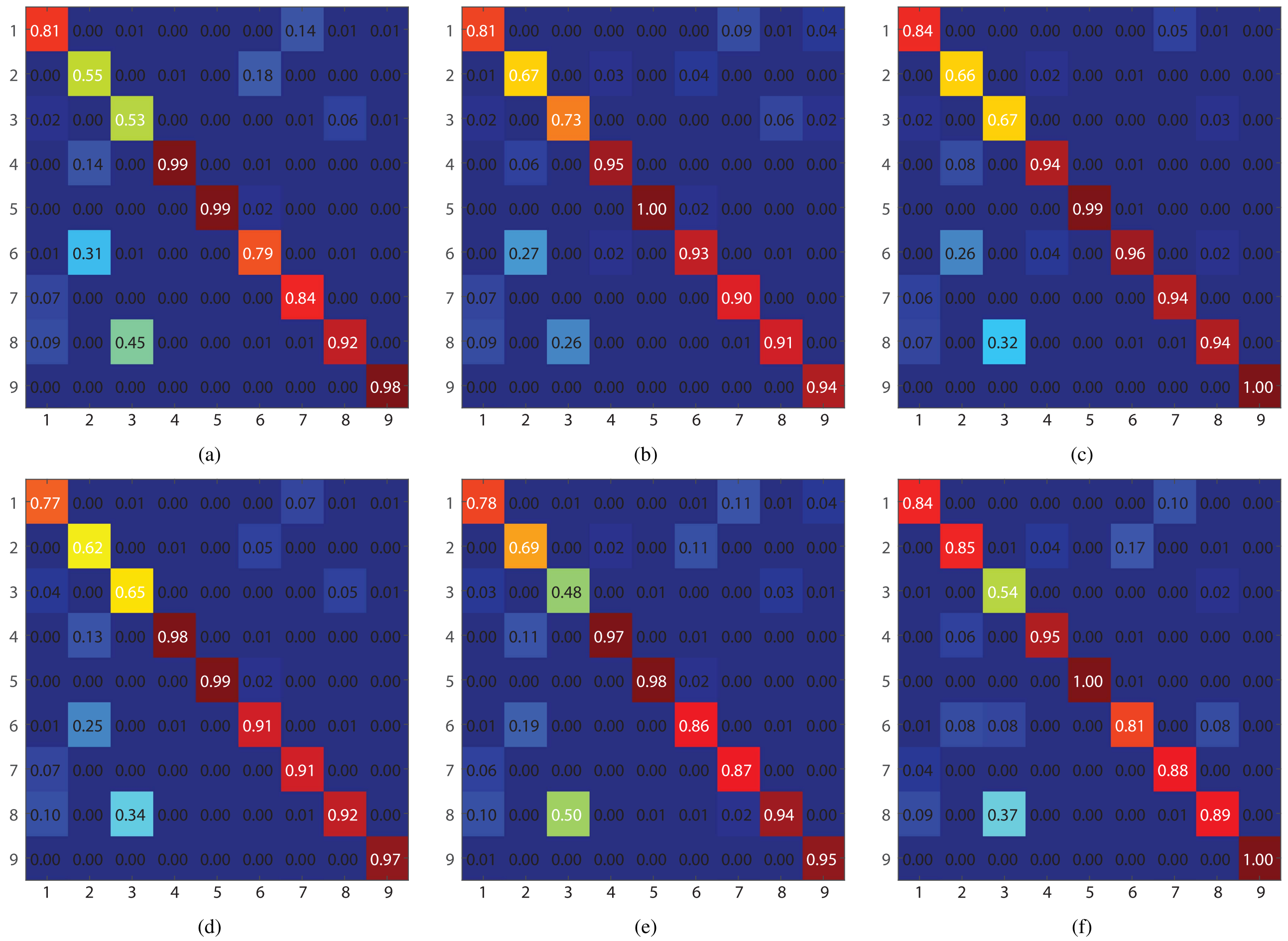


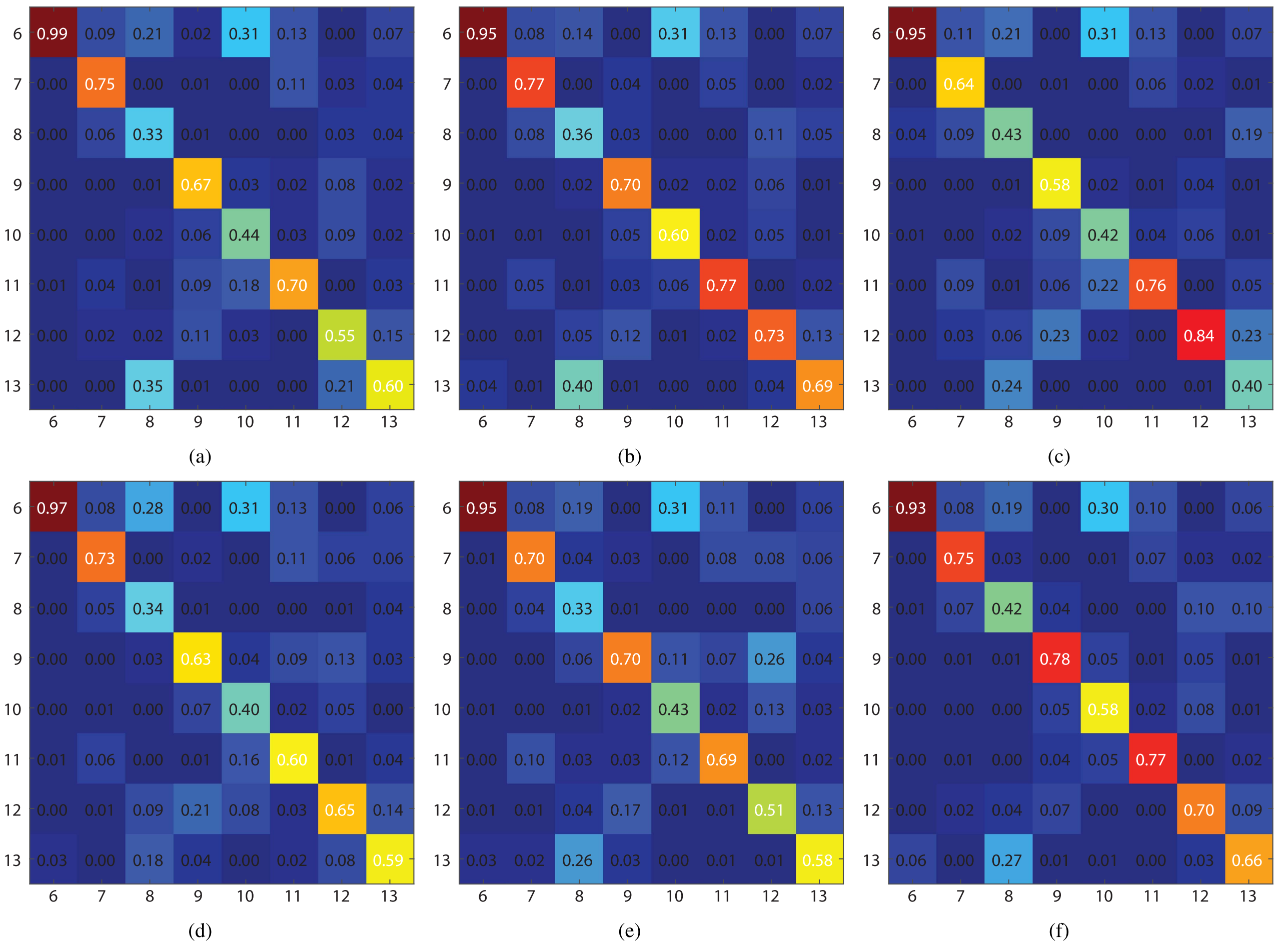
Fig. 8. Confusion matrix of different methods for the Pavia University data set. (a) RF-200. (b) SVM-RBF. (c) CNN. (d) RNN-LSTM. (e) RNN-GRU-tanh. (f) RNN-GRU-PRetanh.

perspective of a hyperspectral pixel is compared with the traditional vector-based models, such as random forest, SVM, and 1-D CNN. In the last part, we discuss processing time.

C. Analysis of the Network

1) Comparisons Between ReLU, tanh, and PRetanh: The activation function is a basic building block of a neural network, because it enables the network to detect nonlinear features in the data. Here, we investigate and compare the behaviors of three activation functions, ReLU, tanh, and PRetanh. Fig. 5 compares the convergence performance of RNN-GRU-ReLU, RNN-GRU-tanh, and RNN-GRU-PRetanh on both the Pavia University and Houston data. All activation functions can make the recurrent network converge except ReLU. Moreover, compared with tanh, the proposed PRetanh activation function starts reducing error earlier and finally reduces the loss to a lower value, which means that the network can converge to a better solution. In particular, PRetanh can obtain the error value of 0.272 on the Pavia University data set after 100 epochs, while the traditional tanh activation function can only achieve 0.334. As Fig. 5(a) shows, the RNN with ReLU as the activation function falls into divergence, which means that we cannot obtain a valid network. For the Houston data set, the recurrent network with the proposed PRetanh can quickly converge to the error of 0.401 after 100 iterations. In the same conditions, tanh can only yield 0.603. ReLU, however, cannot cause the recurrent network to converge. We also compare the classification accuracies of RNN-GRU-tanh and RNN-GRU-PRetanh. As shown in Tables IV–VI, compared with tanh, the network with the proposed PRetanh activation function increases the accuracy significantly by 8.15% of OA, 2.74% of AA, and 0.0847 of the Kappa coefficient, respectively, on the Pavia University data set. For the Houston data set, the accuracy increments on OA, AA, and Kappa coefficient are 4.12%, 6.38%, and 0.0821, respectively. On the Indian Pines data set, our network is able to achieve the accuracy increments of 2.92%, 5.37%, and 0.0733 for OA, AA, and the Kappa coefficient, respectively.

2) Comparison of Recurrent Unit Architecture: The most prominent trait shared between LSTM and GRU is that there is an additive loop of their update from $k-1$ to k, which is lacking in the conventional feedforward neural networks, such as CNNs. In contrast, compared with the traditional recurrent unit like (2), both LSTM and GRU keep the existing content and add the new content on top of it [see (7) and (24)]. These two units, however, have a number of differences as well. LSTM uses three gates and a cell, an input gate, an output



(a)

	6	7	8	9	10	11	12	13
6	0.99	0.09	0.21	0.02	0.31	0.13	0.00	0.07
7	0.00	0.75	0.00	0.01	0.00	0.11	0.03	0.04
8	0.00	0.06	0.33	0.01	0.00	0.00	0.03	0.04
9	0.00	0.00	0.01	0.67	0.03	0.02	0.08	0.02
10	0.00	0.00	0.02	0.06	0.44	0.03	0.09	0.02
11	0.01	0.04	0.01	0.09	0.18	0.70	0.00	0.03
12	0.00	0.02	0.02	0.11	0.03	0.00	0.55	0.15
13	0.00	0.00	0.35	0.01	0.00	0.00	0.21	0.60

(b)

	6	7	8	9	10	11	12	13
6	0.95	0.08	0.14	0.00	0.31	0.13	0.00	0.07
7	0.00	0.77	0.00	0.04	0.00	0.05	0.00	0.02
8	0.00	0.08	0.36	0.03	0.00	0.00	0.11	0.05
9	0.00	0.00	0.02	0.70	0.02	0.02	0.06	0.01
10	0.01	0.01	0.01	0.05	0.60	0.02	0.05	0.01
11	0.00	0.05	0.01	0.03	0.06	0.77	0.00	0.02
12	0.00	0.01	0.05	0.12	0.01	0.02	0.73	0.13
13	0.04	0.01	0.40	0.01	0.00	0.00	0.04	0.69

(c)

	6	7	8	9	10	11	12	13
6	0.95	0.11	0.21	0.00	0.31	0.13	0.00	0.07
7	0.00	0.64	0.00	0.01	0.00	0.06	0.02	0.01
8	0.04	0.09	0.43	0.00	0.00	0.00	0.01	0.19
9	0.00	0.00	0.01	0.58	0.02	0.01	0.04	0.01
10	0.01	0.00	0.02	0.09	0.42	0.04	0.06	0.01
11	0.00	0.09	0.01	0.06	0.22	0.76	0.00	0.05
12	0.00	0.03	0.06	0.23	0.02	0.00	0.84	0.23
13	0.00	0.00	0.24	0.00	0.00	0.00	0.01	0.40

(d)

	6	7	8	9	10	11	12	13
6	0.97	0.08	0.28	0.00	0.31	0.13	0.00	0.06
7	0.00	0.73	0.00	0.02	0.00	0.11	0.06	0.06
8	0.00	0.05	0.34	0.01	0.00	0.00	0.01	0.04
9	0.00	0.00	0.03	0.63	0.04	0.09	0.13	0.03
10	0.00	0.01	0.00	0.07	0.40	0.02	0.05	0.00
11	0.01	0.06	0.00	0.01	0.16	0.60	0.01	0.04
12	0.00	0.01	0.09	0.21	0.08	0.03	0.65	0.14
13	0.03	0.00	0.18	0.04	0.00	0.02	0.08	0.59

(e)

	6	7	8	9	10	11	12	13
6	0.95	0.08	0.19	0.00	0.31	0.11	0.00	0.06
7	0.01	0.70	0.04	0.03	0.00	0.08	0.08	0.06
8	0.00	0.04	0.33	0.01	0.00	0.00	0.00	0.06
9	0.00	0.00	0.06	0.70	0.11	0.07	0.26	0.04
10	0.01	0.00	0.01	0.02	0.43	0.02	0.13	0.03
11	0.00	0.10	0.03	0.03	0.12	0.69	0.00	0.02
12	0.01	0.01	0.04	0.17	0.01	0.01	0.51	0.13
13	0.03	0.02	0.26	0.03	0.00	0.01	0.01	0.58

(f)

	6	7	8	9	10	11	12	13
6	0.93	0.08	0.19	0.00	0.30	0.10	0.00	0.06
7	0.00	0.75	0.03	0.00	0.01	0.07	0.03	0.02
8	0.01	0.07	0.42	0.04	0.00	0.00	0.10	0.10
9	0.00	0.01	0.01	0.78	0.05	0.01	0.05	0.01
10	0.00	0.00	0.00	0.05	0.58	0.02	0.08	0.01
11	0.00	0.01	0.00	0.04	0.05	0.77	0.00	0.02
12	0.00	0.02	0.04	0.07	0.00	0.00	0.70	0.09
13	0.06	0.00	0.27	0.01	0.01	0.00	0.03	0.66

Fig. 9. Zoomed-in view confusion matrix of different methods for the Houston data set. (a) RF-200. (b) SVM-RBF. (c) CNN. (d) RNN-LSTM. (e) RNN-GRU-tanh. (f) RNN-GRU-PRetanh. To show the result more clearly, we show only class #6 to class #13, which are easily misclassified in the Houston data set.

TABLE VII

NUMBER OF TOTAL TRAINABLE PARAMETERS IN DIFFERENT RECURRENT LAYERS

Data set	LSTM	GRU-tanh	GRU-PRetanh
Pavia University (64 units)	16.50 K	12.38 K	12.57 K
Houston (128 units)	65.00 K	48.75 K	49.13 K
Indian Pines (128 units)	79.27 K	63.02 K	63.15 K

gate, a forget gate, and a memory cell, to control the exposure of memory content, while the GRU only employs two gates to control the information flow. In this way, the total number of parameters in the GRU is reduced by about 25%, which makes it the recurrent unit of choice in the recurrent layer for the hyperspectral image classification task. Table VII shows the number of total trainable parameters in different recurrent layers.

Tables IV–VI list the results obtained by our experiments. For all three data sets, the RNN-GRU-PRetanh outperforms the LSTM-based network (RNN-LSTM) on all indexes. Specifically, the RNN-GRU-PRetanh increases the accuracy significantly by 10.86% of OA, 0.45% of AA, and 0.1020 of Kappa, respectively, on the Pavia University data set; by 4.44% of OA, 5.84% of AA, and 0.0717 of the Kappa coefficient, respectively, on the Houston data set; and by 8.11% of OA, 6.61% of AA, and 0.0994 of the Kappa coefficient, respectively, on the Indian Pines data set.

D. Vector-Based Methods Versus Our Recurrent Network

The classification maps of the Pavia University data set obtained by the conventional vector-based models and our network are shown in Fig. 6, and the corresponding accuracy indexes are presented in Table IV. An analysis of the classification accuracies indicates that the SVM with RBF kernel (SVM-RBF) outperforms the random forest model, mainly because the kernel SVM generally handles nonlinear inputs more efficiently than the random forest model. It can be seen that the proposed recurrent network RNN-GRU-PRetanh outperforms the SVM-RBF and CNN in terms of OA and the Kappa coefficient. Compared with SVM-RBF and CNN, the proposed RNN-GRU-PRetanh increases the OA by 10.03% and 8.34%, respectively. Moreover, the proposed network achieves the best accuracies on some specific classes of the Pavia University data, such as asphalt, meadows, metal sheets, and shadows. For instance, the accuracy of the meadows category obtained by RNN-GRU-PRetanh reaches 85.24%,

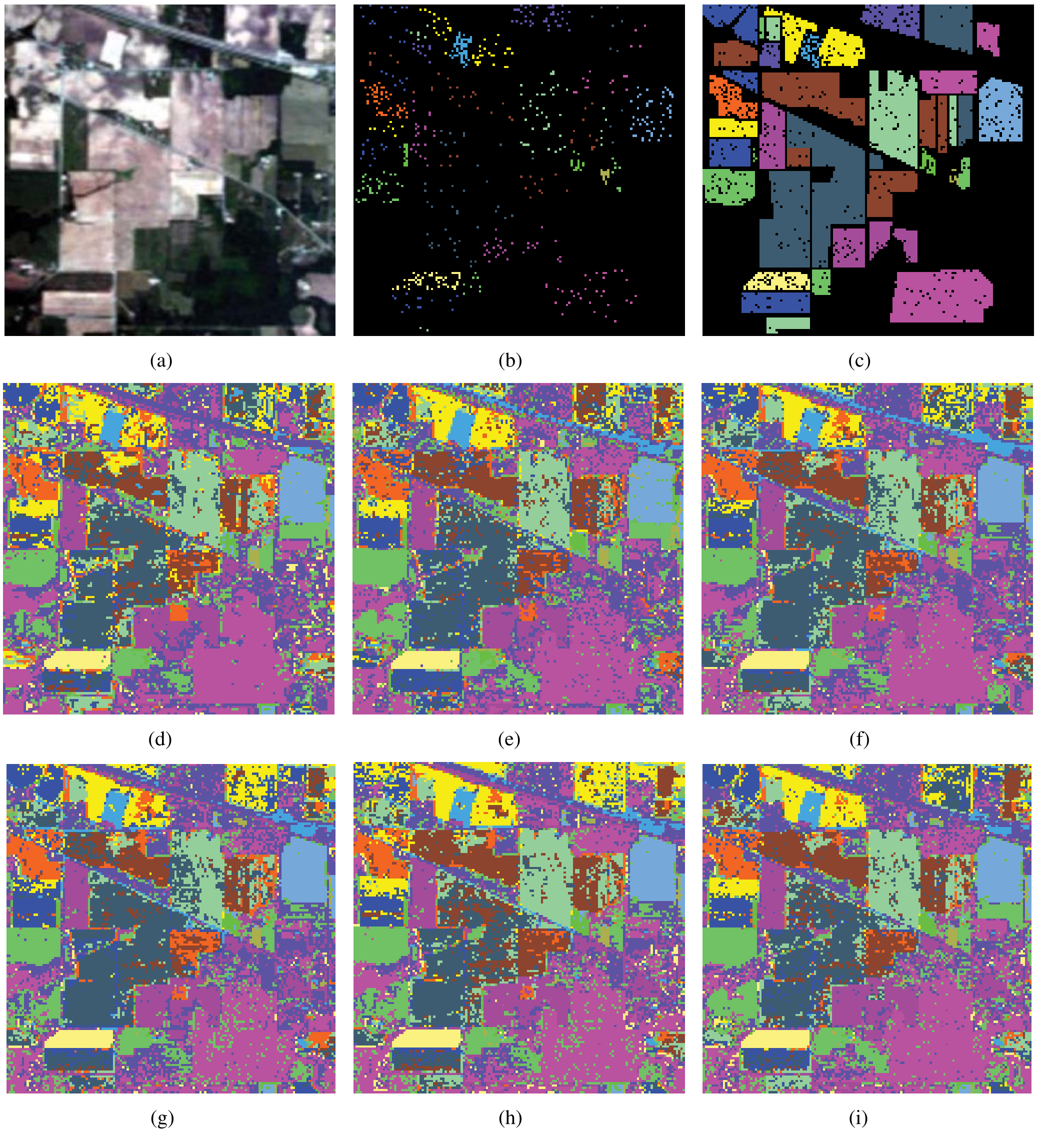
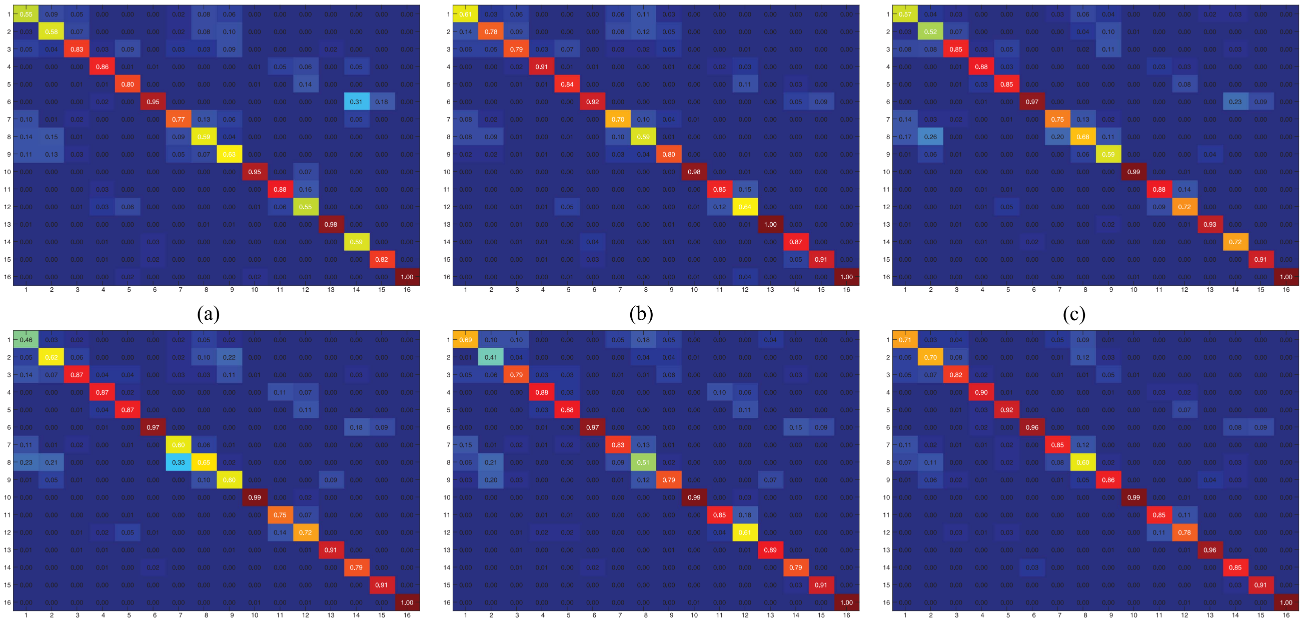


Fig. 10. Classification results obtained by different methods for the Indian Pines scene. (a) True-color composite (bands R: 26, G: 14, and B: 8). (b) Training data. (c) Ground-truth reference. (d) RF-200. (e) SVM-RBF. (f) CNN. (g) RNN-LSTM. (h) RNN-GRU-tanh. (i) RNN-GRU-PRetanh.

and the proposed network can achieve almost 100% on the shadows class.

Fig. 7 shows the classification maps on the Houston data set; the comparison of accuracies between the random forest, SVM-RBF, and RNN-GRU-PRetanh can be found in Table V. The proposed RNN-GRU-PRetanh achieves significantly better scores for OA, AA, and the Kappa coefficient compared with all other methods. Misclassification in this data set lies in similar objects, such as Road-Highway-Railway and Grass Healthy-Grass Stressed-Grass Synthetic. The proposed RNN-GRU-PRetanh achieves the best AA of 70.89% on Road-Highway-Railway, as well as the best AA of 88.63% on Grass Healthy-Grass Stressed-Grass Synthetic. Confusion matrices for the Pavia University data set and the Houston data set can be found in Figs. 8 and 9, respectively. Note that, for the Houston data set, because of the relatively large number of classes, only selected materials that have high misclassification rates are illustrated. In general, the proposed RNN-GRU-PRetanh also tends to show superior performance in distinguishing similar materials.

The classification maps and accuracy assessment for the Indian Pines data set are shown in Fig. 10 and Table VI. It can be seen that the proposed RNN-GRU-PRetanh yields substantially more accurate results than the other methods. Specifically, compared with SVM-RBF and CNN, the improvements in OA achieved by the proposed recurrent network are 15.85% and 4.45%, respectively, and the increments of AA obtained by RNN-GRU-PRetanh are 2.87% and 5.18%, respectively. Fig. 11 also shows that SVM-RBF and CNN are not very effective for discrimination between similar classes such as Grass-Pasture and Grass-Pasture-Mowed because of their similar spectral reflectance. The classification


(a) (b) (c)

(d) (e) (f)

Fig. 11. Confusion matrix of different methods for the Indian Pines data set. (a) RF-200. (b) SVM-RBF. (c) CNN. (d) RNN-LSTM. (e) RNN-GRU-tanh. (f) RNN-GRU-PRetanh.

TABLE VIII
STATISTICS OF TRAINING TIME (min)

Data set	RF-200	SVM-RBF	CNN	Ours
Pavia University	1.2	17.0	33.3	77.4
Houston	0.9	15.6	39.3	88.8
Indian Pines	0.4	1.0	8.2	19.9

TABLE IX
STATISTICS OF TESTING EFFICIENCY (pixels/s)

Methods	RF-200	SVM-RBF	CNN	Ours
EFFICIENCY	2,042.43	2,034.96	9,427.27	8,396.76

of these similar land covers is improved with the proposed recurrent network.

E. Processing Time

Processing time of different methods is compared in this section. All the experiments are conducted on a personal computer equipped with an Intel Core I5 with 2.20 GHz. The training times of different approaches are shown in Table VIII. It is not surprising that deep neural network-based methods, including CNN and RNN, require a longer training time compared with other traditional vector-based classification models, such as random forest and SVM. Fortunately, such differences remain within one to two orders of magnitude. Between CNN and RNN, RNN requires more yet a tolerable training time, as it involves additional channel-by-channel updates. However, one advantage of deep neural networks is that they are fast in testing (see Table IX), which is very important in practice. Also, thanks to the rapid development of hardware technology, especially of GPU, deep neural networks' drawback of a long training time is becoming less and less decisive.

V. CONCLUSION

In this paper, we propose a novel RNN model for hyperspectral image classification, inspired by our observation that hyperspectral pixels can be regarded as sequential data. Specifically, we proposed a newly designed activation function PRetanh for hyperspectral data processing in RNN, providing an opportunity to use fairly high learning rates without the risk of getting stuck in the divergence. Furthermore, a modified GRU with PRetanh was developed to effectively analyze hyperspectral data. For hyperspectral image classification, our proposed recurrent network was shown to provide statistically higher accuracy than SVM-RBF and CNN. The proposed model considers the intrinsic sequential data structure of a hyperspectral pixel for the first time, representing a novel methodology for better understanding, modeling, and processing of hyperspectral data.

In the future, further experiments will be conducted to fully substantiate the features of deep RNN for hyperspectral image processing, providing more accurate analysis for remote sensing applications, such as transfer learning for remote sensing big data analysis and change detection. In addition, this paper only concentrates on modeling hyperspectral pixels in the spectral domain. An end-to-end convolutional-RNN will be considered for spatial-spectral hyperspectral image classification in the future. We believe that such a spatial-spectral network architecture can further improve classification accuracy.

Acknowledgment

The authors would like to thank the National Center for Airborne Laser Mapping for providing the Houston data set. The authors would also thank Prof. P. Gamba from the University of Pavia, Pavia, Italy, for providing the reflective optics system imaging spectrometer data and corresponding reference information.

References

[1] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Boston, MA, USA: Artech House, 2015.

[2] P. Ghamisi, M. D. Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.

[3] Y. Gu, T. Liu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3235–3247, Jun. 2016.

[4] J. Li, M. Khodadadzadeh, A. Plaza, X. Jia, and J. M. Bioucas-Dias, "A discontinuity preserving relaxation scheme for spectral–spatial hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 625–639, Feb. 2016.

[5] P. Ghamisi, J. A. Benediktsson, and J. R. Sveinsson, "Automatic spectral–spatial classification framework based on attribute profiles and supervised feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5771–5782, Sep. 2014.

[6] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2858–2874, May 2014.

[7] A. Ertürk, M.-D. Iordache, and A. Plaza, "Sparse unmixing-based change detection for multitemporal hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 708–719, Feb. 2015.

[8] B. Demir, F. Bovolo, and L. Bruzzone, "Updating land-cover maps by classification of image time series: A novel change-detection-driven transfer learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 300–312, Jan. 2013.

[9] J. Meola, M. T. Eismann, R. L. Moses, and J. N. Ash, "Application of model-based change detection to airborne VNIR/SWIR hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3693–3706, Oct. 2012.

[10] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.

[11] Q. Zhu, Y. Zhong, B. Zhao, G.-S. Xia, and L. Zhang, "Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 747–751, Jun. 2016.

[12] X. Li, L. Mou, and X. Lu, "Scene parsing from an MAP perspective," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1876–1886, Sep. 2015.

[13] X. Lu, X. Li, and L. Mou, "Semi-supervised multitask learning for scene recognition," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1967–1976, Sep. 2015.

[14] L. G. Olmanson, P. L. Brezonik, and M. E. Bauer, "Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: The Mississippi River and its tributaries in Minnesota," *Remote Sens. Environ.*, vol. 130, pp. 254–265, Mar. 2013.

[15] M. S. Moran, Y. Inoue, and E. M. Barnes, "Opportunities and limitations for image-based remote sensing in precision crop management," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 319–346, Sep. 1997.

[16] S. Delalieux, B. Somers, B. Haest, T. Spanhove, J. V. Borre, and C. A. Mücher, "Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers," *Remote Sens. Environ.*, vol. 126, pp. 222–231, Nov. 2012.

[17] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[18] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[19] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[20] J. A. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2000, pp. 813–815.

[21] J. Li, J. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.

[22] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[23] P. Ghamisi, "Spectral and spatial classification of hyperspectral data," Ph.D. dissertation, Dept. Faculty Elect. Comput. Eng., Univ. Iceland, Reykjavik, Iceland, 2015.

[24] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[26] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Aug. 2015, pp. 4346–4354.

[27] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[28] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[29] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[30] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 258619-1–258619-12, Jan. 2015.

[31] K. Makantasis, K. Karantzalos, A. D. Doulamis, and N. D. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2015, pp. 4959–4962.

[32] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[33] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[34] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.

[35] P. Rodriguez, J. Wiles, and J. L. Elman, "A recurrent neural network that learns to count," *Connection Sci.*, vol. 11, no. 1, pp. 5–40, 1999.

[36] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, p. 506, 2016.

[37] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[38] D. Bahdanau, K. Cho, and Y. Bengio. (2014). *Neural Machine Translation by Jointly Learning to Align and Translate*. [Online]. Available: arXiv:1409.0473

[39] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[40] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] A. Graves. (2013). "Generating sequences with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1308.0850

[43] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul./Aug. 2005.

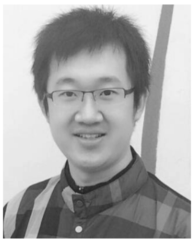
[44] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. (2014). "On the properties of neural machine translation: Encoder-decoder approaches." [Online]. Available: https://arxiv.org/abs/1409.1259

[45] Y. Gal and Z. Ghahramani. (2016). "A theoretically grounded application of dropout in recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1512.05287

[46] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2625–2634.

[47] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. (2016). "Generating images with recurrent adversarial networks." [Online]. Available: https://arxiv.org/abs/1602.05110

[48] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.



Lichao Mou (S'16) received the bachelor's degree in automation from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2012, and the master's degree in signal and information processing from the University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the German Aerospace Center, Wessling, Germany, and the Technical University of Munich, Munich, Germany.

In 2015, he was with the Computer Vision Group, Albert Ludwigs University of Freiburg, Freiburg im Breisgau, Germany. His research interests include remote sensing, computer vision, and machine learning, especially spaceborne remote sensing video analysis and deep networks with their applications in remote sensing.

Mr. Mou was a recipient of the first place award in the 2016 IEEE GRSS Data Fusion Contest.

Pedram Ghamisi (S'12–M'15) received the B.Sc. degree in civil (survey) engineering from Islamic Azad University South Tehran Branch, Tehran, Iran, the M.E. degree (Hons.) in remote sensing from the K. N. Toosi University of Technology, Tehran, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2015.

He was a Post-Doctoral Research Fellow with the University of Iceland. Since 2015, he has been a Post-Doctoral Research Fellow with Signal Processing in Earth Observation, Technical University of Munich, Munich, Germany. He has been with GIScience and 3-D spatial data processing with the Institute of Geography, Heidelberg University, Heidelberg, Germany, since 2015. He has also been a Researcher with the Remote Sensing Technology Institute, German Aerospace Center, Wessling, Germany, on deep learning, since 2015. His research interests include machine learning, deep learning, hyperspectral image analysis, and multisensor data fusion.

Dr. Ghamisi received the Best Researcher Award for M.Sc. students from the K. N. Toosi University of Technology from 2010 to 2011. At the 2013 IEEE International Geoscience and Remote Sensing Symposium, Melbourne, he received the IEEE Mikio Takagi Prize for winning the Student Paper Competition at the conference among almost 70 people. He received the prestigious Alexander von Humboldt Fellowship in 2015. He was selected as a Talented International Researcher by the Iran's National Elites Foundation in 2016.

Xiao Xiang Zhu (S'10–M'12–SM'14) received the bachelor's degree in space engineering from the National University of Defense Technology, Changsha, China, in 2006, and the M.Sc., Dr.-Ing., and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

Since 2011, she has been a Scientist with the Remote Sensing Technology Institute, German Aerospace Center, Wessling, Germany, where she is currently the Head of the Team Signal Analysis. Since 2013, she has also been a Helmholtz Young Investigator Group Leader, where she was also appointed as a TUM Junior Fellow. In 2015, she was appointed as the Professor for Signal Processing in Earth Observation with TUM. She was a Guest Scientist or a Visiting Professor with the Italian National Research Council, Naples, Italy, Fudan University, Shanghai, China, The University of Tokyo, Tokyo, Japan, and the University of California at Los Angeles, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. Her research interests include advanced InSAR techniques, such as high dimensional tomographic SAR imaging and SqueeSAR; computer vision in remote sensing, including object reconstruction and multidimensional data visualization; and big data analysis in remote sensing, and modern signal processing, including innovative algorithms, such as sparse reconstruction, nonlocal means filter, robust estimation, and deep learning, with applications in the field of remote sensing, such as multi/hyperspectral image analysis.

Dr. Zhu is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Corrections to “Deep Recurrent Neural Networks for Hyperspectral Image Classification”

Lichao Mou, *Student Member, IEEE*, Pedram Ghamisi, *Member, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

Here, we correct some errors caused by a programming bug (a data type error) in overall accuracies (OAs) reported in [1]. The corrected OAs are underlined and shown in bold in Tables I–III.

In addition, we also correct the OAs in the text as follows.

1) Page 3650, right column, line 9: 8.15% of OA should be changed to 7.29% of OA.
2) Page 3650, right column, line 12: 4.12% should be corrected to 7.67%.
3) Page 3650, right column, line 14: 2.92% should be corrected to 6.53%.
4) Page 3651, left column, line 13: 10.86% of OA should be changed to 9.07% of OA.
5) Page 3651, right column, line 1: 4.44% of OA should be corrected to 6.74% of OA.
6) Page 3651, right column, line 2: 8.11% of OA should be changed to 8.62% of OA.
7) Page 3651, right column, line 17&18: 10.03% and 8.34% should be changed to 6.17% and 5.72%.
8) Page 3652, right column, line 11: 15.85% and 4.45% should be corrected to 3.89% and 4.27%.

References

[1] L. Mou, P. Ghamisi, and X. X. Zhu, “Deep recurrent neural networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017, doi: 10.1109/TGRS.2016.2636241.

TABLE I

THE CLASSIFICATION ACCURACIES OF DIFFERENT TECHNIQUES IN PERCENTAGES FOR PAVIA UNIVERSITY

Class No.	Class Name	RF-200	SVM-RBF	CNN	RNN-LSTM	RNN-GRU-tanh	RNN-GRU-PRetanh
1	Asphalt	80.85	80.80	83.73	77.45	78.42	84.45
2	Meadows	55.29	66.78	65.70	61.83	69.17	85.24
3	Gravel	52.93	73.18	67.03	64.60	47.83	54.31
4	Trees	98.79	95.17	94.03	97.98	97.16	95.17
5	Metal Sheets	99.26	99.55	99.41	99.18	97.84	99.93
6	Bare Soil	78.76	92.90	96.30	91.19	85.86	80.99
7	Bitumen	84.36	90.08	93.83	90.90	86.84	88.35
8	Bricks	91.58	91.20	93.56	92.29	94.27	88.62
9	Shadows	98.20	93.77	99.79	97.47	94.93	99.89
OA	-	71.37	78.82	**79.27**	**75.92**	**77.70**	**84.99**
AA	-	82.23	87.05	88.15	85.88	83.59	86.33
Kappa	-	0.6484	0.7358	0.7423	0.7028	0.7201	0.8048

Manuscript received August 26, 2016; revised October 29, 2016; accepted November 15, 2016. Date of publication December 25, 2017; date of current version January 26, 2018. This work was supported in part by the China Scholarship Council, in part by the Alexander von Humboldt Foundation, in part by the Helmholtz Association through the framework of the Young Investigators Group SiPEO under Grant VH-NG-1018, and in part by the European Research Council under the European Union’s Horizon 2020 Research and Innovation Programme under Grant ERC-2016-StG-714087 (So2Sat). *(Corresponding author: Xiao Xiang Zhu.)*

The authors are with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Wessling, Germany, and also with Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: lichao.mou@dlr.de; pedram.ghamisi@dlr.de; xiao.zhu@dlr.de).

Digital Object Identifier 10.1109/TGRS.2017.2779459

TABLE II

CLASSIFICATION ACCURACIES OF TEST SAMPLES ON THE HOUSTON DATA SET

Class No.	Class Name	RF-200	SVM-RBF	CNN	RNN-LSTM	RNN-GRU-tahn	RNN-GRU-PRetanh
1	Grass Healthy	82.62	81.96	81.20	81.86	82.43	82.53
2	Grass Stressed	83.36	80.55	83.55	81.20	82.42	83.36
3	Grass Synthetic	98.02	99.80	99.41	99.41	97.23	100
4	Tree	91.76	92.23	91.57	90.06	89.30	90.53
5	Soil	97.06	97.63	94.79	93.09	78.22	97.82
6	Water	99.30	95.10	95.10	96.50	95.10	93.01
7	Residential	75.37	76.59	63.53	73.41	70.43	75.37
8	Commercial	32.95	35.52	42.64	34.09	32.57	42.36
9	Road	67.14	70.44	58.17	62.61	70.25	77.62
10	Highway	43.73	60.04	41.80	39.96	43.24	57.63
11	Railway	70.11	76.57	75.71	60.44	69.07	77.42
12	Parking Lot 1	54.95	73.10	84.15	65.42	50.72	69.74
13	Parking Lot 2	59.65	68.77	40.00	58.95	58.25	66.32
14	Tennis Court	99.19	100	98.79	96.76	97.98	100
15	Running Track	97.67	98.10	97.89	88.37	96.83	95.98
OA	-	72.93	77.09	**74.05**	**71.05**	**70.12**	**77.79**
AA	-	76.86	80.43	76.55	74.81	74.27	80.65
Kappa	-	0.7091	0.7536	0.7200	0.6889	0.6785	0.7606

TABLE III

ACCURACY COMPARISON FOR THE INDIAN PINES DATA SET

Class No.	Class Name	RF-200	SVM-RBF	CNN	RNN-LSTM	RNN-GRU-tahn	RNN-GRU-PRetanh
1	Alfalfa	54.84	60.77	56.79	46.03	68.93	70.59
2	Corn-notill	58.42	77.68	52.17	61.73	40.94	70.28
3	Corn-min	82.61	79.35	85.33	86.96	78.80	81.52
4	Corn	85.91	91.05	87.92	87.02	87.92	90.16
5	Grass-pasture	80.49	84.36	85.22	86.66	87.52	91.97
6	Grass-trees	94.76	92.03	97.49	97.49	97.27	96.13
7	Grass-pasture-mowed	77.34	69.61	74.62	59.69	82.79	84.75
8	Hay-windrowed	59.43	59.31	67.99	64.89	50.58	59.64
9	Oats	63.48	79.61	58.87	60.46	79.43	86.17
10	Soybean-notill	95.06	97.53	98.77	98.77	98.77	99.38
11	Soybean-mintill	88.26	85.21	87.62	75.32	84.73	84.97
12	Soybean-clean	54.85	63.64	72.42	71.82	61.21	77.58
13	Wheat	97.78	100	93.33	91.11	88.89	95.56
14	Woods	58.97	87.18	71.79	79.49	79.49	84.62
15	Buildings-grass-trees	81.82	90.91	90.91	90.91	90.91	90.91
16	Stone-steel-towers	100	100	100	100	100	100
OA	-	69.79	72.78	**72.40**	**68.05**	**70.14**	**76.67**
AA	-	77.13	82.39	80.08	78.65	79.89	85.26
Kappa	-	0.6589	0.6931	0.6852	0.6372	0.6633	0.7366

B Mou L., Ghamisi P., Zhu X., 2018. Unsupervised Spectral-Spatial Feature Learning via Deep Residual Conv-Deconv Network for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, 56(1), 391-406.

Unsupervised Spectral–Spatial Feature Learning via Deep Residual Conv–Deconv Network for Hyperspectral Image Classification

Lichao Mou, *Student Member, IEEE*, Pedram Ghamisi, *Member, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*— Supervised approaches classify input data using a set of representative samples for each class, known as *training samples*. The collection of such samples is expensive and time demanding. Hence, unsupervised feature learning, which has a quick access to arbitrary amounts of unlabeled data, is conceptually of high interest. In this paper, we propose a novel network architecture, fully Conv–Deconv network, for unsupervised spectral–spatial feature learning of hyperspectral images, which is able to be trained in an end-to-end manner. Specifically, our network is based on the so-called encoder–decoder paradigm, i.e., the input 3-D hyperspectral patch is first transformed into a typically lower dimensional space via a convolutional subnetwork (encoder), and then expanded to reproduce the initial data by a deconvolutional subnetwork (decoder). However, during the experiment, we found that such a network is not easy to be optimized. To address this problem, we refine the proposed network architecture by incorporating: 1) residual learning and 2) a new unpooling operation that can use memorized max-pooling indexes. Moreover, to understand the "black box," we make an in-depth study of the learned feature maps in the experimental analysis. A very interesting discovery is that some specific "neurons" in the first residual block of the proposed network own good description power for semantic visual patterns in the object level, which provide an opportunity to achieve "free" object detection. This paper, for the first time in the remote sensing community, proposes an end-to-end fully Conv–Deconv network for unsupervised spectral–spatial feature learning. Moreover, this paper also introduces an in-depth investigation of learned features. Experimental results on two widely used hyperspectral data, Indian Pines and Pavia University, demonstrate competitive performance obtained by the proposed methodology compared with other studied approaches.**

***Index Terms*— Convolutional network, deconvolutional network, hyperspectral image classification, residual learning, unsupervised spectral–spatial feature learning.**

I. Introduction

Along with the development of different earth observation missions, hyperspectral imagery has been accessible at a reasonable cost over the last decade. Since hyperspectral images are characterized in hundreds of continuous observation bands, throughout the electromagnetic spectrum with high spectral resolution, such data have attracted considerable attention in the remote sensing community [1]. On the other hand, the analysis of hyperspectral images is of high importance in many practical applications, such as urban development [2]–[5], monitoring of land changes [6]–[9], and resource management [10], [11]. To benefit from these types of data, supervised hyperspectral image classification is among the most active research areas in hyperspectral analysis.

There is a vast literature on supervised classification models such as decision trees [12], random forests [13], [14], and support vector machines (SVMs) [15], [16]. A random forest [14] is an ensemble learning approach that operates by constructing several decision trees in the training course and outputting the classes of the input hyperspectral pixels via integration of predictions of the individual trees. In contrast, as a significant branch of the supervised machine learning algorithm, SVMs have achieved a great success in various applications due to the fact that they can handle high-dimensional data with a limited number of training samples. SVM works by mapping data to a kernel-included high-dimensional feature space seeking an optimal decision hyperplane that can best separate data samples, when data points are not linearly separable. SVM, therefore, has been considered to be an effective and stable algorithm for hyperspectral image classification task. In addition, some extensions of the SVM model [17], [18] have been proposed for hyperspectral data analysis to improve discrimination capability of the classifier. However, random forests and SVMs are attributed as "shallow" models, which means that their ability to deal with nonlinear data, e.g., hyperspectral data demonstrate dense nonlinearity, is limited compared with the "deep" ones.

With the investigation of hyperspectral image classification, a major finding is that various atmospheric scattering conditions, complicated light scattering mechanisms, interclass similarity, and intraclass variability result in hyperspectral imaging procedure being inherently nonlinear [19]. It is believed that, compared with the "shallow" models,

Manuscript received January 7, 2017; revised May 4, 2017 and June 29, 2017; accepted August 3, 2017. Date of publication October 24, 2017; date of current version December 27, 2017. This work is jointly supported by the China Scholarship Council, Alexander von Humboldt Foundation, the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No [ERC-2016-StG-714087], Acronym: *So2Sat*), and Helmholtz Association under the framework of the Young Investigators Group "SiPEO" (VH-NG-1018, www.sipeo.bgu.tum.de). *(Corresponding author: Xiao Xiang Zhu.)*

The authors are with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Wessling, Germany, and also with Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: lichao.mou@dlr.de; p.ghamisi@gmail.com; xiao.zhu@dlr.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2748160

deep learning architectures are able to extract high-level, hierarchical, and abstract features, which are generally more robust to the nonlinear input data. So far, some studies in the community have focused on making use of deep learning models for hyperspectral image classification. For instance, Chen *et al.* [20] employed a stacked auto-encoder to extract hierarchical features from the spectral domain of hyperspectral images for the purpose of classification. In [21], a restricted Boltzmann machine (RBM) and its extension, deep belief network (DBN), were introduced for the classification of hyperspectral data by learning spectral-based features. Tao *et al.* [22] presented a multiscale sparse stacked auto-encoder to learn an effective feature representation from unlabeled data, and then the learned features were fed into a linear SVM for hyperspectral data classification. Very recently, Mou *et al.* [23] proposed a novel recurrent neural network with a new activation function and a modified gated recurrent unit for hyperspectral image classification, which can effectively analyze hyperspectral pixels as sequential data and then determine information categories via network reasoning.

Most of the aforementioned networks, e.g., auto-encoder, RBM, and DBN, are both early and fairly simple 1-D deep learning architectures totally equipped with fully connected layers. Consequently, there are a lot of trainable parameters that need to be estimated, which is an undesirable case given that available labeled training samples for remote sensing image classification are often limited [24]. Moreover, it should be noted that the processing mechanism of the 1-D networks and the vector-based feature alignment can lead to the loss of structure information for hyperspectral imagery, as it has an intrinsic 2-D data structure in the spatial domain.

Convolutional neural network (CNN), an important branch of the deep learning family, has been attracting attention, due to the fact that they are capable of automatically discovering relevant contextual 2-D spatial features in image categorization tasks. In addition, a CNN makes use of local connections to deal with spatial dependencies via sharing weights, and thus can significantly reduce the number of parameters of the network in comparison with the conventional 1-D fully connected neural networks. CNNs have already outperformed other methodologies in various domains of machine learning and computer vision such as large-scale natural image recognition [25]–[28], object detection [29], [30], and scene interpretation [31]–[35]. Very recently, a few supervised CNN-based models have been proposed for spectral–spatial classification of hyperspectral remote sensing images. Chen *et al.* [36] introduced a supervised ℓ_2 regularized 3-D CNN-based feature extraction model to extract efficient spectral–spatial features for the purpose of classification. Ghamisi *et al.* [19] proposed a self-improving CNN (SICNN) model, which combined a CNN with a fractional order Darwinian particle swarm optimization (FODPSO) algorithm to iteratively select the most informative bands suitable for training the designed CNN. Makantasis *et al.* [37] exploited a CNN to encode spectral and spatial information of input hyperspectral data followed by a multilayer perceptron to conduct the hyperspectral image classification task. Zhao and Du [38] proposed a spectral–spatial feature-based classification framework, which jointly makes use of a local discriminant embedding-based dimension reduction algorithm and a CNN for the purpose of land cover classification. Aptoula *et al.* [39] fed attribute profile features instead of original hyperspectral data into a CNN, which led to a performance improvement.

Those CNNs trained in a supervised manner via backpropagation, which improved the state-of-the-art performance on the hyperspectral image classification task. Despite the big success of the supervised CNNs, they have at least one potential drawback detailed as follows: there is a need for a good supply of labeled training samples to be used for supervised training. However, these are difficult to collect, and there are diminishing returns of making the labeled data set larger and larger. In other words, the supervised CNNs generally suffer from either small number of training samples or imbalanced data sets.

Hence, unsupervised spectral–spatial feature learning, which has a quick access to arbitrary amounts of unlabeled data, is conceptually of high interest. In general, the main purpose of unsupervised feature learning is to extract useful features from unlabeled data, to detect and remove input redundancies, and to preserve only essential aspects of the data in robust and discriminative representations. In a pioneer work moving from the supervised CNN to unsupervised CNN, Romero *et al.* [40] proposed an unsupervised convolutional network for learning spectral–spatial features using sparse learning to estimate the weights of the network. However, this model was trained in a greedy layer-wise fashion, i.e., it is not an end-to-end network. In this paper, we aim to propose an end-to-end network, fully Conv–Deconv network, for unsupervised spectral–spatial feature learning of hyperspectral imagery. Basically, our network architecture is based on the so-called encoder–decoder paradigm. Specifically, the input is first transformed into a typically lower dimensional space via a convolutional subnetwork (encoder), and then expanded to reproduce the initial data by a deconvolutional subnetwork (decoder). Moreover, the trained unsupervised Conv–Deconv network can be adapted to the classification of hyperspectral data by cutting off the deconvolutional subnetwork, replacing the loss function, and fine-tuning it to the new task, i.e., adjusting the weights using backpropagation. With this approach, typically much smaller training sets are sufficient. In detail, our work contributes to the literature in three major aspects.

1) We propose an end-to-end deep Conv–Deconv neural network, which is composed of a convolutional subnetwork and a deconvolutional subnetwork with a specially designed unpooling layer. Learning such a 2-D encoder–decoder-based network for unsupervised spectral–spatial feature learning of hyperspectral data has not been investigated yet to the best of our knowledge.
2) Since our network is fairly deep, it might easily converge to an inappropriate solution if small learning rates are used. On the other hand, simply boosting convergence with high learning rates leads to exploding the gradient problem. In this paper, we resolve this issue by introducing residual learning in our Conv–Deconv network. To the best of our knowledge, this is the first

use of residual learning to train networks for remote sensing data analysis.

3) Our unsupervised network is able to preserve the neighborhood relations and spatial locality of 3-D hyperspectral cubes in its latent high-level feature representations, while the conventional 1-D fully connected unsupervised network architectures such as auto-encoder, RBM, and DBN do not scale well to realistic-sized high-dimensional hyperspectral data in terms of computational complexity.
4) To understand the "black box" of the proposed network, we make an in-depth investigation. We found that some specific "neurons" in the first residual block of the network are capable of precisely capturing semantic visual patterns in object level, which makes it possible to achieve a high-quality unsupervised object detection capability for hyperspectral images.

The rest of this paper is organized as follows. An introduction to the traditional unsupervised network architectures is briefly given in Section II. The details of the proposed fully Conv–Deconv network with residual learning for unsupervised spectral–spatial feature extraction of hyperspectral data are described in Section III. The network setup, network analysis, experimental results, and a comparison with state-of-the-art approaches are provided in Section IV. Finally, Section V concludes this paper.

II. Preliminaries

Several types of traditional 1-D unsupervised network architectures have been leveraged for feature learning of hyperspectral data. In this section, we recall the basic principles of such models.

A. Auto-Encoder

An auto-encoder [41] takes an input $\boldsymbol{x} \in \mathbb{R}^D$ and first maps it to a latent representation $\boldsymbol{h} \in \mathbb{R}^M$ via a nonlinear mapping

$$\boldsymbol{h} = f(\boldsymbol{\Theta x} + \boldsymbol{\beta}) \tag{1}$$

where $\boldsymbol{\Theta}$ is a weight matrix to be estimated during the training course, $\boldsymbol{\beta}$ is a bias vector, and f stands for a nonlinear function such as the logistic sigmoid function and hyperbolic tangent function. The encoded feature representation $\boldsymbol{h}$ is then used to reconstruct the input $\boldsymbol{x}$ by a reverse mapping

$$\boldsymbol{y} = f(\boldsymbol{\Theta}'\boldsymbol{h} + \boldsymbol{\beta}') \tag{2}$$

where $\boldsymbol{\Theta}'$ is usually constrained to be the form of $\boldsymbol{\Theta}' = \boldsymbol{\Theta}^T$, using the same weight for encoding the input and decoding the latent representation. The reconstruction error is defined as the Euclidian distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ that is constrained to approximate the input data $\boldsymbol{x}$, i.e., making $\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \to 0$. The parameters of the auto-encoder are generally optimized by stochastic gradient descent (SGD) [42]. Fig. 1 illustrates the structure of the auto-encoder.

B. Sparse Auto-Encoder

The conventional auto-encoder relies on the dimension of the latent representation $\boldsymbol{h}$ being smaller than that of input $\boldsymbol{x}$, i.e., $M < D$, which means it tends to learn a low-dimensional compressed representation. However, when $M > D$, one can still discover an interesting structure, by enforcing a sparsity constraint on the hidden units. Formally, given a set of unlabeled data $\boldsymbol{X} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$, training a sparse auto-encoder is to find the optimal parameters by minimizing the following loss function:

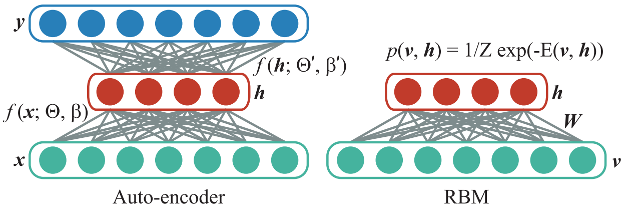


Fig. 1. Two classical unsupervised network architectures. (Left) Auto-encoder. (Right) RBM.

$$\mathbb{E} = \frac{1}{N}\sum_{i=1}^{N}\left(J(\boldsymbol{x}^i, \boldsymbol{y}^i; \boldsymbol{\Theta}, \boldsymbol{\beta}) + \lambda\sum_{j=1}^{M}\mathrm{KL}(\rho\|\hat{\rho}_j)\right) \tag{3}$$

where $J(\boldsymbol{x}^i, \boldsymbol{y}^i; \boldsymbol{\Theta}, \boldsymbol{\beta})$ is an average sum-of-squares error term, which represents the reconstruction error between the input $\boldsymbol{x}^i$ and its reconstruction $\boldsymbol{y}^i$. $\mathrm{KL}(\rho\|\hat{\rho}_j)$ is the Kullback-Leibler (KL) divergence between a Bernoulli random variable with mean ρ and a Bernoulli random variable with mean $\hat{\rho}_j$. KL divergence is a standard function for measuring how similar two distributions are, and it can be described as follows:

$$\mathrm{KL}(\rho\|\hat{\rho}_j) = \rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}. \tag{4}$$

In the sparse auto-encoder model, KL divergence is called sparsity penalty term that provides the sparsity constraint, and λ controls the weight of the sparsity penalty term. Similar to the auto-encoder, the optimization of a sparse auto-encoder can be achieved via the backpropagation and SGD [42].

C. RBM and DBN

Unlike the deterministic network architectures such as auto-encoder or sparse auto-encoder, an RBM is a stochastic undirected graphical model consisted of a visible layer and a hidden layer, and it has symmetric connections between these two layers, and no connecting exists within the hidden layer or the input layer. The energy function of an RBM can be defined as follows:

$$E(\boldsymbol{x}, \boldsymbol{h}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} - (\boldsymbol{h}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{h}) \tag{5}$$

where $\boldsymbol{W}$, $\boldsymbol{c}$, and $\boldsymbol{b}$ are the weights of the RBM model. The joint probability distribution of the RBM is defined as

$$p(\boldsymbol{x}, \boldsymbol{h}) = \frac{1}{Z}\exp(-\mathbb{E}(\boldsymbol{x}, \boldsymbol{h})) \tag{6}$$

where Z is a normalization constant. The form of the RBM makes the conditional probability distribution computationally feasible, when $\boldsymbol{x}$ or $\boldsymbol{h}$ is fixed. The structure of the RBM is depicted in Fig. 1.

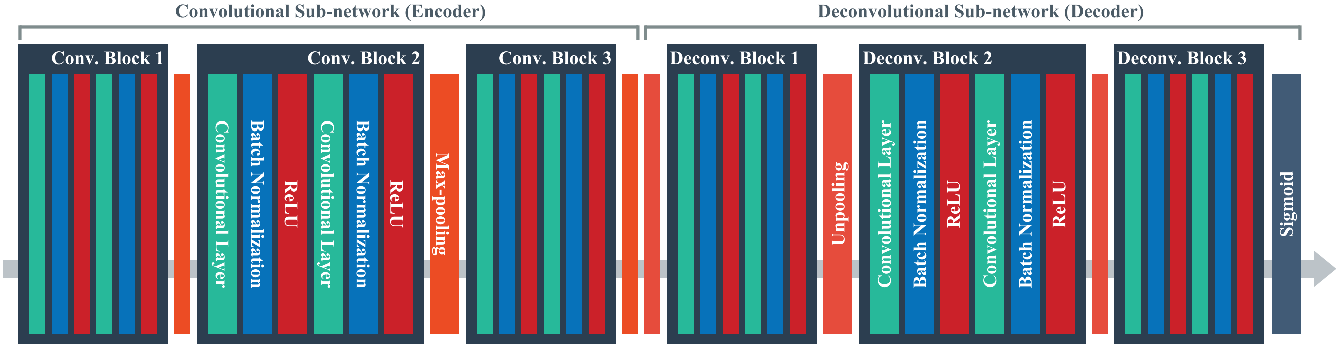


Fig. 2. We propose a network architecture that learns to extract spectral–spatial features by reconstructing the initial input 3-D hyperspectral patches, being trained end to end. There are no fully connected layers, and hence it is a fully Conv–Deconv network. The proposed network architecture is composed of two parts, i.e., convolutional subnetwork and deconvolutional subnetwork. The former corresponds to an encoder that transforms the input 3-D hyperspectral patches to abstract feature representations, whereas the latter plays the role of decoder that reproduces the initial input data from the encoded features. Each layer in the convolutional subnetwork has a corresponding decoder layer in the deconvolutional subnetwork.

The feature representation ability of a single RBM is limited. However, its real power emerges when a couple of RBMs are stacked, forming a DBN [43]. Hinton *et al.* [43] proposed a greedy approach that trains RBM in each layer to efficiently train a DBN.

III. Methodology

CNNs have shown to be very successful on a range of visual recognition tasks [25]–[27], [29]–[33]. Such tasks, in common, can be posed as discriminative supervised learning problems, and hence, can be resolved by CNNs, which are well known to be effective at learning input–output relations given an adequate number of labeled data sets. Normally, a task solved by making use of CNNs involves learning mappings from concrete raw images to some sort of condensed abstract outputs, such as category. Here, we are interested in training an end-to-end neural network to extract features in an unsupervised fashion, which means we need to leverage a network to solve a concrete-to-concrete problem instead of the traditional concrete-to-abstract one. This brings up a question in mind: what is a good network architecture for our purpose?

A. Initial Conv–Deconv Network Architecture

1) Analysis and Modeling: Denote by $(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y})$ random variables representing a 3-D hyperspectral patch, its encoded feature representation, and the reconstructed output. The joint probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$ can be described as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x}) \tag{7}$$

where $p(\boldsymbol{x})$ is the distribution of 3-D hyperspectral patches and $p(\boldsymbol{y}|\boldsymbol{x})$ is the distribution of reconstructed outputs given the hyperspectral patches. Thus, the conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ can be written as

$$p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y}, \boldsymbol{h}|\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{h})p(\boldsymbol{h}|\boldsymbol{x}) \tag{8}$$

where $p(\boldsymbol{h}|\boldsymbol{x})$ indicates the distribution of the encoded feature representations given the input hyperspectral patches. As a special case, $\boldsymbol{y}$ may be a deterministic function of $\boldsymbol{x}$. Ideally, we would like to find $p(\boldsymbol{h}|\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{h})$, but direct application of Bayesian theory is not feasible. We, therefore, in this paper resort to an estimate function $f(\boldsymbol{x})$ that minimizes the following mean squared error objective:

$$\mathbb{E}_{\boldsymbol{x}}\|\boldsymbol{x} - f(\boldsymbol{x})\|_2^2. \tag{9}$$

The minimizer of this loss is the conditional expectation

$$\hat{f}(\boldsymbol{x}_0) = \mathbb{E}_{\boldsymbol{y}}[\boldsymbol{y}|\boldsymbol{h}] + \mathbb{E}_{\boldsymbol{h}}[\boldsymbol{h}|\boldsymbol{x} = \boldsymbol{x}_0] \tag{10}$$

that is the expected reconstructed output given a hyperspectral patch.

Given a set of unlabeled 3-D hyperspectral patches $\{\boldsymbol{x}_i\}$, we learn the weights $\boldsymbol{\Theta}$ of a network $f(\boldsymbol{x}; \boldsymbol{\Theta})$ to minimize a Monte Carlo estimate of the loss (9)

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_i \|\boldsymbol{x}_i - f(\boldsymbol{x}_i; \boldsymbol{\Theta})\|_2^2. \tag{11}$$

This means that we train the network to reproduce the input results in learning high-level abstract features in an unsupervised manner.

In this paper, we propose a fully Conv–Deconv network (see Fig. 2) in which the desired output is the input data itself. The proposed network architecture is composed of two parts, i.e., the convolutional subnetwork and deconvolutional subnetwork. The former corresponds to an encoder that transforms the input 3-D hyperspectral patch $\boldsymbol{x}_i$ to abstract feature representation $\boldsymbol{h}_i$, whereas the latter plays the role of a decoder that reproduces the initial input data from the encoded feature. Each layer in the convolutional subnetwork has a corresponding decoder layer in the deconvolutional subnetwork.

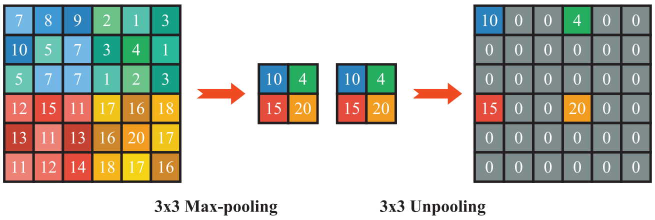
2) Convolutional Subnetwork: The design of the architecture of the convolutional subnetwork is mainly inspired by the philosophy of the VGG Nets [26]. The input hyperspectral patch is fed into a stack of convolutional layers, where we leverage convolutional filters with a very small receptive field of 3 × 3, rather than making use of larger ones, such as 5 × 5 or 7 × 7. The reason is that 3 × 3 convolutional filters are the smallest kernels to seize patterns in different directions, such as center, up/down, and left/right, but still have an advantage: the usage of small convolutional filters will increase the nonlinearities inside the network and thus make the network more discriminative.

In addition, the convolutional stride in the convolutional subnetwork is fixed to 1 pixel; the spatial padding of convolutional layer input is such that the spatial resolution of feature maps is preserved after convolution, in other words, the padding is 1 pixel for the used 3 × 3 convolutional layers. Spatial pooling is achieved by carrying out several max-pooling layers, which follow some of the convolutional layers. In particular, max pooling is performed over 3 × 3 pixel windows with stride 3.

In a nutshell, the convolutional layers in the convolutional subnetwork consist of 3 × 3 filters and follow the following two rules: 1) the convolutional layers in each convolutional block are with the same feature map size and have the same number of filters and 2) the number of channels of the feature maps increases in the deeper convolutional blocks, roughly doubling after each max-pooling layer, which is meant to preserve the time complexity per layer as far as possible. All layers in the convolutional subnetwork are equipped with a rectified linear unit (ReLU) [25] as activation function. ReLU is one of several keys to the recent success of deep neural networks and can be defined as $f(x) = \max(0, x)$. Compared with the conventional activation functions, such as sigmoid and hyperbolic tangent function, the usage of ReLU can expedite convergence of the training course and result in better solutions.

3) Deconvolutional Subnetwork: The convolutional subnetwork is in charge of extracting high-level abstract spectral–spatial feature representation of the input 3-D hyperspectral patch, by interleaving convolutional layers and max-pooling layers, i.e., spatially shrinking the feature maps layer by layer. Pooling is necessary to allow agglomerating information over large areas of feature maps and, more fundamentally, to make the network computationally feasible. However, pooling leads to reduced resolution of the feature maps; hence, in order to reconstruct the initial input data, we need to find a way to refine this coarse pooled representation.

Our approach to this refinement is to construct a deconvolutional subnetwork. The main ingredient is deconvolutional operation, which performs reverse operation of the convolutional subnetwork and reconstructs the original input data from the abstract feature representation. The deconvolutional operation consists of unpooling and convolution. In order to map the encoded feature to a high-dimensional hyperspectral cube, we need unpooling to unpool the feature maps, i.e., to increase their spatial span, as opposed to the pooling (spatially shrinking the feature maps) implemented by the convolutional subnetwork. More specifically, the unpooling [44], [45] is performed by simply replacing each entry of a feature map by an $s \times s$ block with the entry value in the top-left corner and zeros elsewhere (see Fig. 3). With this operation, the height and the width of the feature maps are increased s times. In this network, we made use of $s = 3$, as the size of the receptive field in the max-pooling layers of the convolutional subnetwork is 3 × 3. When a convolutional block is preceded by an unpooling layer, we can thus think of the combination of unpooling and convolutional block as the inverse operation of "convolutional block + pooling" performed in the convolutional subnetwork.



7	8	9	2	1	3
10	5	7	3	4	1
5	7	7	1	2	3
12	15	11	17	16	18
13	11	13	16	20	17
11	12	14	18	17	16

10	4
15	20

10	4
15	20

10	0	0	4	0	0
0	0	0	0	0	0
0	0	0	0	0	0
15	0	0	20	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Fig. 3. Illustration of (Left) max pooling and (Right) unpooling as used in the fully Conv–Deconv network described in Section III-A.

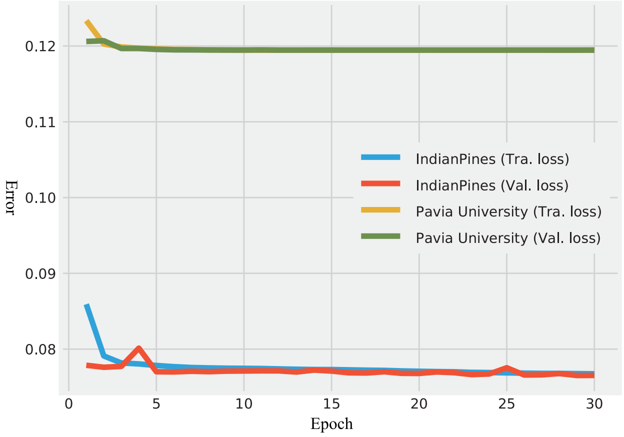


Fig. 4. Learning curves for the initial fully Conv–Deconv network on the Indian Pines data set and the Pavia University data set. Although the network starts greatly reducing errors on both the training and validation samples during the first few epochs, it rapidly converges to a fairly high value, which means the learning of the network is significantly slowed down and eventually gets stuck into a local minimum. This indicates that such a network architecture is not easy to optimize.

The configuration of convolutional blocks in the deconvolutional subnetwork is the same with the convolutional subnetwork, namely, 3 × 3 receptive field, 1 pixel padding, and ReLU as activation function.

B. Refined Network Architecture

1) Difficulty of Training Conv–Deconv Network: In Section III-A, we have systematically built a reasonable network architecture for our task, but a problem will arise when we attempt to train the network. As can be seen in Fig. 4, although the network starts greatly reducing errors on both the training and validation samples during the first few epochs, it rapidly converges to a fairly high value, which means the learning of the network is significantly slowed down and eventually gets stuck into a local minimum. This indicates that such network architecture is not easy to optimize. We think the obstacles to train the proposed fully Conv–Deconv network are as follows.

1) In the Conv–Deconv network, the exact copy of the input high-dimensional 3-D hyperspectral patch has to go through all layers until it reaches the output layer. With many weight layers, this becomes an end-to-end relation requiring very long-term memory. For this reason, the notorious vanishing gradient problem [46], [47] can

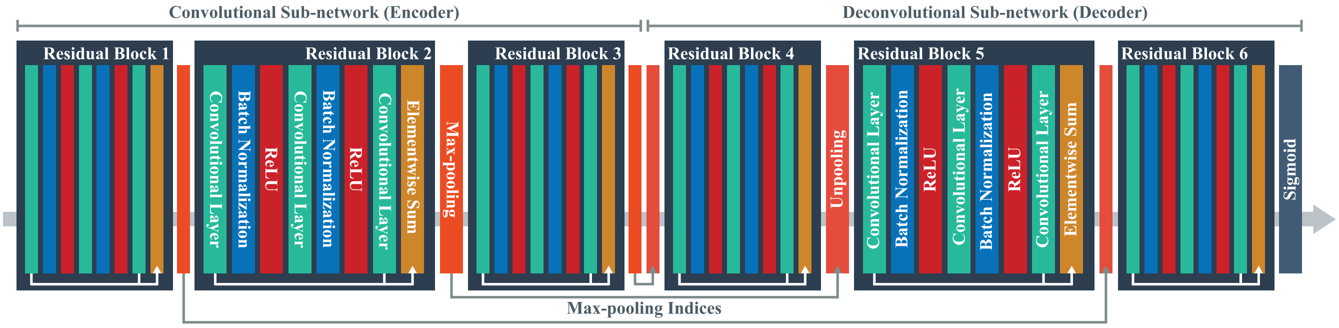


Fig. 5. We refine the proposed fully Conv–Deconv network architecture by incorporating residual learning and a more appropriate unpooling operation, which can use memorized max-pooling indices from the corresponding encoded feature maps and enables reconstruction to be more accurate.

be critical, which handicaps the learning process of the network.

2) The unpooling operation [44], [45] in the deconvolutional subnetwork increases the spatial resolution of feature maps by simply adding zeros, which ignores the location of the maximum value in the receptive field of pooling layer, leading to loss of edge information during the decoding procedure. Without this detailed information, it is difficult for the optimizer to lead the network to better solutions.

To address the aforementioned problems, in this section, we refine the proposed fully Conv–Deconv network architecture by incorporating residual learning and a new unpooling operation that can use memorized max-pooling indices from the corresponding encoded feature maps and enables reconstruction to be more accurate. The new network architecture is shown in Fig. 5.

2) Conv–Deconv Network With Residual Learning: Residual learning has recently shown appealing performance in the concrete-to-abstract deep network architectures on many challenging visual tasks, such as image classification [27], [48] and object detection [27]. One main merit offered using the residual learning is that it helps in handling the vanishing gradient problem and degradation problem [27]. In this paper, we are interested in introducing the residual learning to the proposed concrete-to-concrete Conv–Deconv network in order to resolve the network training problem.

The proposed Conv–Deconv network with residual learning is a modularized network architecture that stacks residual blocks. Similar to the convolutional blocks, a residual block consists of several convolutional layers that are with the same feature map size and have the same number of filters. However, it performs the following calculation:

$$\varphi_l = g(\boldsymbol{\phi}_l) + \mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l) \tag{12}$$

$$\boldsymbol{\phi}_{l+1} = f(\varphi_l). \tag{13}$$

Here, $\boldsymbol{\phi}_l$ indicates the feature maps that are fed into the lth residual block and satisfies $\boldsymbol{\phi}_0 = \boldsymbol{x}$ where $\boldsymbol{x}$ is the input 3-D hyperspectral patch. $\boldsymbol{\Theta}_l = \{\boldsymbol{\Theta}_{l,k} | 1 \leq k \leq K\}$ represents a collection of weights associated with the lth residual block, and K denotes that there are K convolutional layers in a residual block. Moreover, $\mathcal{F}$ is the residual function and is generally achieved by few stacked convolutional layers, e.g., a convolutional block described in Section III-A. The function f indicates the activation function such as a linear activation function or ReLU, and f works after element-wise addition. The function g is fixed to an identity mapping: $g(\boldsymbol{\phi}_l) = \boldsymbol{\phi}_l$.

If f adopts a linear activation function and also acts as an identity mapping, i.e., $\boldsymbol{\phi}_{l+1} = \varphi_l$, we can obtain the output of the lth residual block by putting (12) into (13)

$$\boldsymbol{\phi}_{l+1} = \boldsymbol{\phi}_l + \mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l)\,. \tag{14}$$

In contrast, a convolutional block only performs the following computation:

$$\boldsymbol{\phi}_{l+1} = \mathcal{H}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l). \tag{15}$$

Recursively, like

$$\begin{aligned}\boldsymbol{\phi}_{l+2} &= \boldsymbol{\phi}_{l+1} + \mathcal{F}(\boldsymbol{\phi}_{l+1}; \boldsymbol{\Theta}_{l+1})\\ &= \boldsymbol{\phi}_l + \mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l) + \mathcal{F}(\boldsymbol{\phi}_{l+1}; \boldsymbol{\Theta}_{l+1})\end{aligned} \tag{16}$$

we will get the following recurrence formula:

$$\boldsymbol{\phi}_L = \boldsymbol{\phi}_l + \sum_{i=l}^{L-1} \mathcal{F}(\boldsymbol{\phi}_i; \boldsymbol{\Theta}_i) \tag{17}$$

for any shallower block l and any deeper block L.

As exhibited in (17), the network with residual learning has some interesting and nice properties.

1) The feature maps $\boldsymbol{\phi}_L$ of any deeper residual block L can be considered to be adding the feature maps $\boldsymbol{\phi}_l$ of any shallower block l and a residual function in a form of $\sum_{i=1}^{L-1} \mathcal{F}$, representing that the network is in a residual fashion and is capable of learning some new features between any blocks l and L.

2) With both the g and f being identity mappings, i.e., $g(\boldsymbol{\phi}_l) = \boldsymbol{\phi}_l$ and $f(\varphi_l) = \varphi_l$, a network with residual learning creates a direct path for propagating information through the entire network, which can effectively avoid the vanishing gradient problem.

These two respects are in contrast to the Conv–Deconv network equipped with common convolutional blocks (see Section III-A) in which the feature maps $\boldsymbol{\phi}_L$ are a set of matrix products, namely, $\prod_{i=0}^{L-1} \boldsymbol{\Theta}_i \boldsymbol{\phi}_0$.

The content discussed above illustrates the forward propagation procedure of the Conv–Deconv network with residual

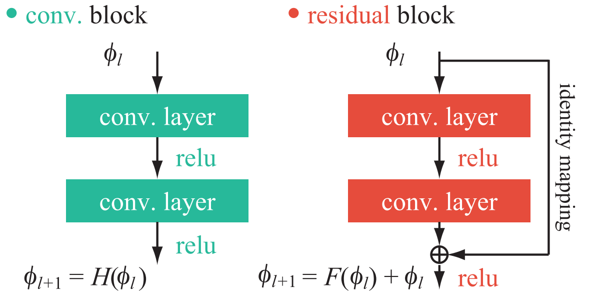
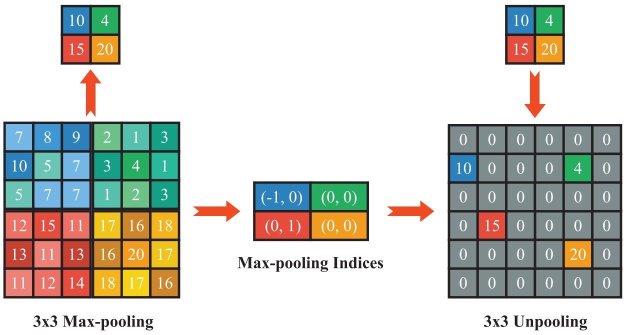


Fig. 6. Comparison between the convolutional block and the residual block. Here, $\boldsymbol{\phi}_l$ denotes the input and $\boldsymbol{\phi}_{l+1}$ is any desired output. The convolutional block hopes that two convolutional layers are able to fit $\boldsymbol{\phi}_{l+1}$ by directly learning a mapping $\mathcal{H}$. In contrast, the two convolutional layers are expected to learn a residual function $\mathcal{F}$ to let $\boldsymbol{\phi}_{l+1} = \mathcal{F}(\boldsymbol{\phi}_l) + \boldsymbol{\phi}_l$ in the residual block.



7	8	9	2	1	3
10	5	7	3	4	1
5	7	7	1	2	3
12	15	11	17	16	18
13	11	13	16	20	17
11	12	14	18	17	16

10	4
15	20

(-1, 0)	(0, 0)
(0, 1)	(0, 0)

0	0	0	0	0	0
10	0	0	0	4	0
0	0	0	0	0	0
0	15	0	0	0	0
0	0	0	0	20	0
0	0	0	0	0	0

Fig. 7. Illustration of the unpooling operation in the refined Conv–Deconv network (see Section III-B), using max-pooling indices that are capable of recording the location of the maximum value in each local pooling region during pooling in the convolutional subnetwork.

learning. However, how the residual learning can help us to effectively train the proposed deep network? To answer this question, we need to dive into the backward propagation process. Denoted by $\mathbb{E}$ indicating the loss function, according to the chain rule of backpropagation, we can obtain

$$\frac{\partial \mathbb{E}}{\partial \boldsymbol{\phi}_l} = \frac{\partial \mathbb{E}}{\partial \boldsymbol{\phi}_L}\frac{\partial \boldsymbol{\phi}_L}{\partial \boldsymbol{\phi}_l} = \frac{\partial \mathbb{E}}{\partial \boldsymbol{\phi}_L}\left(1+\frac{\partial}{\partial \boldsymbol{\phi}_l}\sum_{i=l}^{L-1}\mathcal{F}(\boldsymbol{\phi}_i;\boldsymbol{\Theta}_i)\right). \tag{18}$$

Equation (18) implies that the gradient $(\partial\mathbb{E}/\partial\boldsymbol{\phi}_l)$ can be decomposed into two additive terms: a term of $(\partial\mathbb{E}/\partial\boldsymbol{\phi}_L)$ that directly propagates information without concerning any weight layers and another term of $(\partial\mathbb{E}/\partial\boldsymbol{\phi}_L)((\partial/\partial\boldsymbol{\phi}_l)\sum_{i=l}^{L-1}\mathcal{F})$ that propagates through the weight layers. The former term ensures that the information can be propagated back to any shallower residual block l directly. In addition, since $(\partial/\partial\boldsymbol{\phi}_l)\sum_{i=l}^{L-1}\mathcal{F}$ basically cannot always be -1 for all training data in a batch, it is almost impossible that (18) is canceled out for a minibatch. This implies that the gradient information of a layer in the network does not vanish even while the trainable weights are arbitrarily small, which is the key to make the deep network feasible for the purpose of training and to answer the question mentioned above. Given the activation function of the last layer is sigmoid, on the contrary, the initial Conv–Deconv network easily suffers from the vanishing gradient problem, which leads the learning procedure is slowed down or even stopped. Fig. 6 shows a comparison between the convolutional block [Fig. 6 (left)] and the residual block [Fig. 6 (right)].

3) More Accurate Unpooling: To acquire more appropriate unpooled feature maps and more precise reconstruction output, the max-pooling indices computed in the max-pooling layers of the corresponding encoder can be used to perform nonlinear upsampling of the feature maps. And, reusing the max-pooling indices in the deconvolutional subnetwork has several practical merits, including that it is able to improve boundary delineation and eliminates the need for learning to upsample. The unpooled feature maps produced by this form of unpooling are sparse. Then, the unpooled feature maps are convolved with trainable filters to generate dense feature maps.

Goroshin *et al.* [49] recently presented a soft version of max and arg max operations that can take not only the maximum value in the receptive field of a max-pooling layer but also the corresponding index of that value. In particular, these two operations can be computed as follows:

$$\mu = \sum_{\mathcal{V}} z(i,j)\frac{\exp(\alpha z(i,j))}{\sum_{\mathcal{V}}\exp(\alpha z(i,j))} \approx \max_{\mathcal{V}} z(i,j) \tag{19}$$

$$\nu = \sum_{\mathcal{V}} [i,j]^T\frac{\exp(\alpha z(i,j))}{\sum_{\mathcal{V}}\exp(\alpha z(i,j))} \approx \arg\max_{\mathcal{V}} z(i,j) \tag{20}$$

where (i, j) stands for the spatial location index in the receptive field of a max-pooling layer and takes normalized values from -1 to 1, and $z(i, j)$ presents the value of the given location on a feature map. $\mathcal{V}$ is the receptive field. Note that α is a hyperparameter that controls soft pooling such that the lager the α, the closer the soft pooling approaches max pooling. With the max and arg max operations, the max-poling indices can be obtained in every pooling layer.

Then we make use of interpolation in the unpooling layers of the deconvolutional subnetwork by handling the values conveyed by the max-pooling indices (see Fig. 7). The use of max-pooling indices enables location information to be more accurately represented and thus enables the feature maps to capture fine details about the input 3-D hyperspectral patch.

C. Usage of Learned Features for Classification by Fine-Tuning the Network

Once the Conv–Deconv network is trained, the convolutional subnetwork, i.e., the encoder, can be regarded as an effective feature extractor. The key idea, here, is that the internal layers of the convolutional subnetwork can act as a generic extractor of spectral–spatial representation, which, first, can be trained by learning an identity mapping in the encoder–decoder architecture and then reused on other target tasks like classification. With this fine-tuning, we do not have to use a large number of labeled data to train a valid network for the purpose of supervised classification. In contrast, taking into consideration the fact that the total number of trainable parameters of a deep 2-D convolutional network is huge, a direct learning of so many parameters from the limited number of training samples is problematic. For fine-tuning, we cut off the deconvolutional subnetwork, introduce a new fully connected layer with softmax as a classifier, and fine-tune this

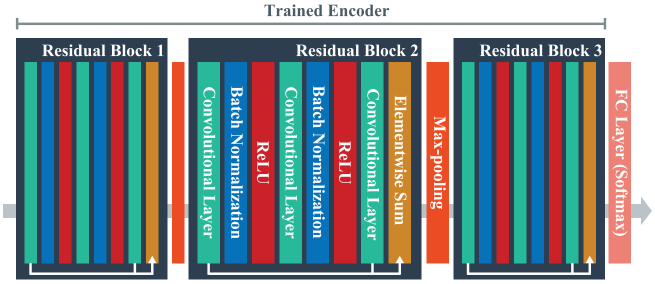


Fig. 8. Illustration of fine-tuning.

TABLE I
NUMBER OF TRAINING AND TEST SAMPLES USED IN THE INDIAN PINES DATA SET

Class No.	Class Name	Training	Test
1	Alfalfa	50	1384
2	Corn-notill	50	784
3	Corn-min	50	184
4	Corn	50	447
5	Grass-pasture	50	697
6	Grass-trees	50	439
7	Grass-pasture-mowed	50	918
8	Hay-windrowed	50	2418
9	Oats	50	564
10	Soybean-notill	50	162
11	Soybean-mintill	50	1244
12	Soybean-clean	50	330
13	Wheat	50	45
14	Woods	15	39
15	Buildings-grass-trees	15	11
16	Stone-steel-towers	15	5
	TOTAL	695	9671

new layer with limited labeled training samples, making the network significantly easier to be trained for the classification task. Fig. 8 illustrates this process.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Data Description

1) Indian Pines: This data set was acquired over the Indian Pines agricultural site in northwestern Indiana. It was collected with an airborne visible/infrared imaging spectrometer (AVIRIS) sensor in June 1992. The AVIRIS sensor comprises 220 spectral channels ranging from 400 to 2500 nm. In this data set, 20 bands affected by atmosphere absorption have been removed, and the remaining 200 spectral bands are investigated in this paper. The data set consists of 145 × 145 pixels, and the spatial resolution is 20 m/pixel. The available training samples of this data set cover 16 classes of interests. Table I provides information about different classes and their corresponding training and test samples.

2) Pavia University: The second data set was captured by reflective optics system imaging spectrometer (ROSIS) covering the Engineering School at the University of Pavia, and presents nine classes, mostly related to land covers. The image is of 610 × 340 pixels with a spatial resolution of 1.3 m/pixel and was collected under the HySens project managed by the German Aerospace Center. The hyperspectral imagery consists of 115 spectral channels ranging from 430 to 860 nm. In this paper, we made use of 103 spectral channels, after removing 12 noisy bands. Table II provides information about all nine classes of this data set with their corresponding training and test samples.

TABLE II
NUMBER OF TRAINING AND TEST SAMPLES USED IN THE PAVIA UNIVERSITY DATA SET

Class No.	Class Name	Training	Test
1	Asphalt	548	6631
2	Meadows	540	18649
3	Gravel	392	2099
4	Trees	524	3064
5	Metal sheets	265	1345
6	Bare Soil	532	5029
7	Bitumen	375	1330
8	Bricks	514	3682
9	Shadows	231	947
	TOTAL	3921	42776

B. General Information

To evaluate the performance of different approaches for hyperspectral image classification, the following evaluation criteria are used.

1) *Overall Accuracy (OA):* This measure represents the number of samples that are classified correctly, divided by the number of test samples.
2) *Average Accuracy (AA):* This index shows the average value of the classification accuracies of all categories.
3) *Kappa Coefficient:* This metric is a statistical measurement that provides information regarding the amount of agreement between the ground truth map and the final classification map. It is the percentage agreement corrected by the level of agreement, which could be expected due to the chance alone. In general, it is considered to be a more robust index than a simple percent agreement calculation, since k takes into account the agreement occurring by chance [1].

In addition, in order to evaluate the significance of the classification accuracies obtained by different approaches, a statistical test is conducted. Since the samples that were used for two different classification approaches are not independent, we evaluate the significance of two classification results with McNemar's test, which is given by [50]

$$z_{12} = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}}$$

where f_{ij} is the number of correctly classified samples in classification i and incorrectly in classification j. McNemar's test is based on the standardized normal test statistic and therefore, the null hypothesis, which is "no significant difference," is rejected at the widely used $p = 0.05$ ($|z| > 1.96$) level of significance.

To validate the effectiveness of the proposed network architecture for the purpose of hyperspectral image classification, the novel classification method is compared with the most widely used supervised models, random forest [13], [14] and SVMs [15], [16]. In addition, in this paper, the experiments making use of other supervised deep learning methods such as 1-D CNN and 2-D CNN are also carried out to verify the validity of the proposed network. The approaches included in the comparison are summarized as follows.

1) *RF-200:* Random forest with 200 trees.
2) *SVM-RBF:* SVMs with an RBF kernel are implemented using the libsvm package.[1] Furthermore, fivefold cross-validation is taken into account to tune the hyperplane parameters.
3) *1-D CNN:* The network architecture of the 1-D CNN is designed as in [51] and includes an input layer, convolutional layer, max-pooling layer, fully connected layer, and output layer. The number of the convolutional filters is 20 for all data sets. The length of each convolutional filter and the pooling size are 11 and 3, respectively. Moreover, 100 hidden units are contained in the fully connected layer.
4) *2-D CNN:* We follow the architecture of the 2-D CNN as used in [36]. It contains three convolutional layers that are equipped with 4×4, 5×5, and 4×4 convolutional filters, respectively. The convolutional layers—apart from the last one—are followed by the max-pooling layers. In addition, the numbers of the convolutional filters for the convolutional layers are 32, 64, and 128, respectively.
5) *SICNN:* An SICNN model solves the curse of dimensionality and the lack of available training samples by iteratively selecting the most informative bands suitable for the designed network via FODPSO [19].
6) *Initial Conv–Deconv Network:* The fully Conv–Deconv network with the plain convolutional blocks and the unpooling operation implemented in [44] and [45] (see Section III-A).
7) *Residual Conv–Deconv Network:* Our final network architecture makes use of the residual blocks and a more accurate unpooling operation. Section III-B shows the details.

Note that, to make the proposed approach fully comparable with other supervised classifiers in the literature, we used the standard sets of training and test samples for the data sets.

The fully Conv–Deconv network was trained using the Adam algorithm [52], and all the suggested default parameters were used for all the following experiments. The number of convolutional filters increases toward deeper layers of the convolutional subnetworks: 64 for the first residual block, 128 for the following block, and 256 for the last one. This rule is turned over for the deconvolutional subnetwork. All the convolutional layers are with ReLU as nonlinear activation function except the last layer that uses sigmoid activation. All weight matrices in the network and bias vectors are initialized with a uniform distribution, and the values of these weight matrices and bias vectors are initialized in the range $[-0.1, 0.1]$. The number of unlabeled data samples used for training the Conv–Deconv network on both Indian Pines and Pavia University is 10 000. These unlabeled samples are randomly selected from the whole images. Prior to training the Conv–Deconv network, we normalize the hyperspectral data in the range of 0–1. Then, all the weights can be updated during the training procedure. Once the training of Conv–Deconv network is complete, we can start to fine-tune the network for hyperspectral data classification. We made use of SGD with a fairly low learning rate of 0.0001 in order to fine-tune the network. For fine-tuning, in both hyperspectral data sets, we randomly chose 10% of the training samples as the validation set. That is, during fine-tuning, we used 90% of the training samples to learn the parameters and the remaining 10% of the training samples as validation to tune the superparameters, such as the numbers of convolutional filters in the convolutional layers. All test samples are used to evaluate the final performance of the learned spectral–spatial feature representations and the fine-tuned network for classification.

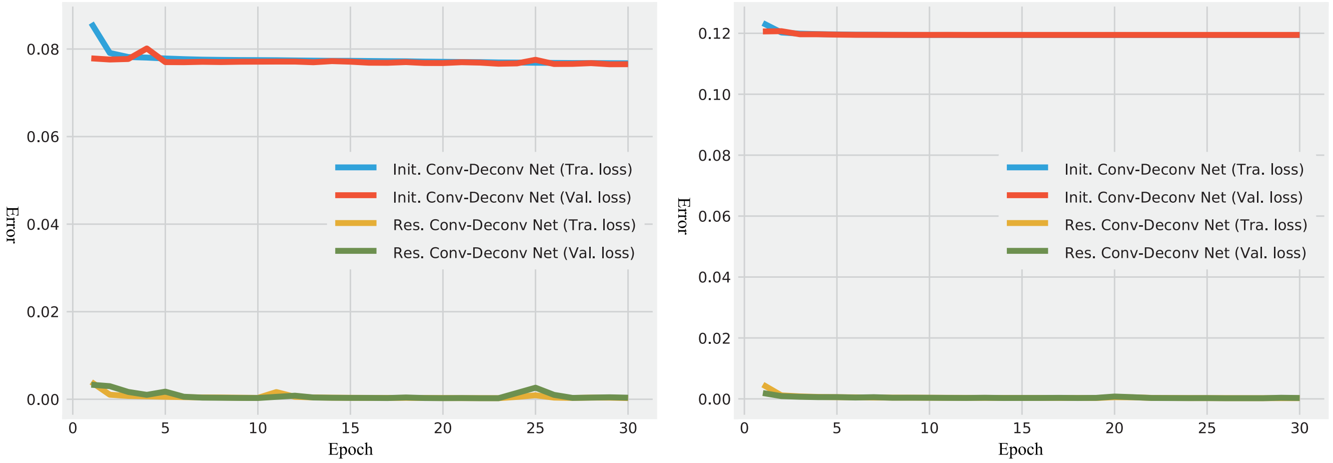
The experiments are organized into three parts. The first part aims primarily at evaluating the learning procedures of the initial Conv–Deconv network and the residual Conv–Deconv network. Moreover, the learned feature maps are also shown and discussed in this part. In the second part, the effectiveness of the proposed network is compared with other state-of-the-art models such as random forest, SVM, 1-D CNN, and 2-D CNN. In the last part, we comment on the processing time.

C. Analysis of the Conv–Deconv Networks

1) Learning Curves: We first investigate the behavior of the initial Conv–Deconv network and the residual Conv–Deconv network during the training process, before we present the performance of the networks for the classification task. The qualities of the trained networks can be reflected by learning curves. As shown in Fig. 9, the initial Conv–Deconv network starts reducing error earlier on both the training samples and the validation samples but finally reduces the loss to a relatively high value, which means the learning of the network is apparently slowed down and the network converges to a local minimum in the end. In contrast, with residual learning, the residual Conv–Deconv network shows strong convergence ability. In particular, the residual Conv–Deconv network can obtain the training error value of 0.000276 on the Indian Pines data set after 30 epochs, while the initial Conv–Deconv network can achieve only 0.0767. For the Pavia University data set, the residual Conv–Deconv network can quickly converge to the error of 0.000238 after 30 iterations. In the same condition, the initial Conv–Deconv network can yield only 0.120. Furthermore, since we do not observe the overfitting problem in Fig. 9, the trained residual Conv–Deconv network can be thought as a good model for the follow-up fine-tuning stage.

2) Feature Visualization and Analysis: In order to understand the "black box" of the Conv–Deconv network, we show and analyze the learned feature maps. Specifically, we use the Pavia University data set to perform an in-depth study of the learned feature representation. Note that we do not have any fully connected layer in the residual Conv–Deconv network, which allows the trained network to take hyperspectral images of arbitrary size as input. Fig. 10 shows feature visualizations from the first residual block of the residual Conv–Deconv

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvm/



(a)

(b)

Fig. 9. Learning curves for the initial Conv–Deconv network and the residual Conv–Deconv network on the training samples and the validation samples of (a) Indian Pines data set and (b) Pavia University data set. With residual learning and the new unpooling operation, we can lead the network to a better solution. Here, we use the Adam optimizer with a default learning rate of 0.001.

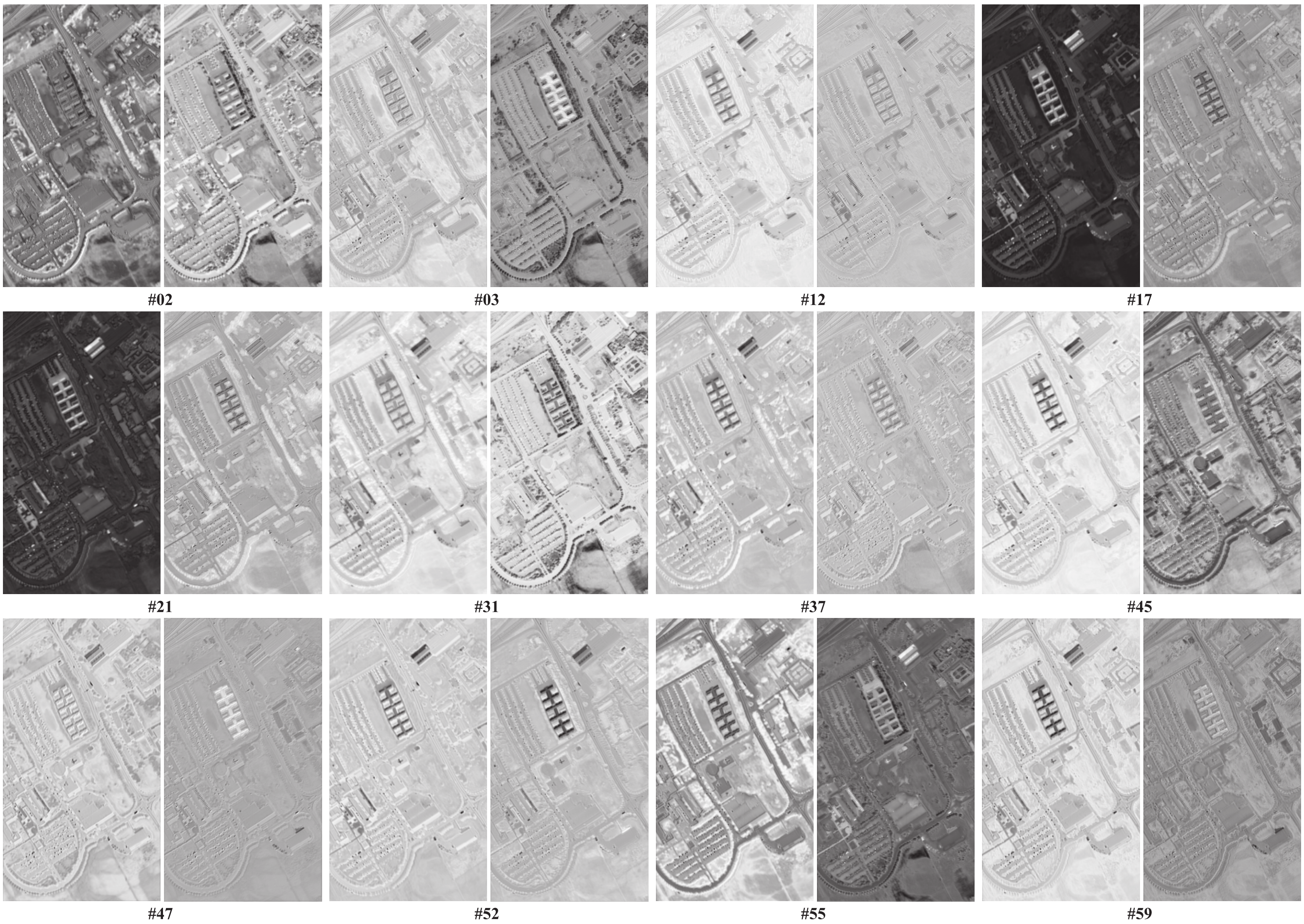
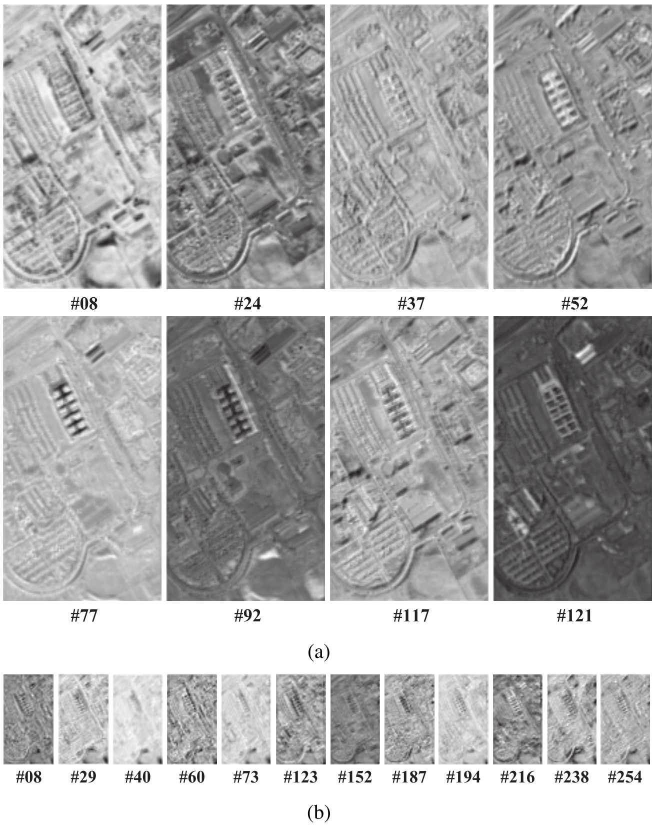


Fig. 10. Feature visualizations from the first residual block of the residual Conv–Deconv network once training is complete on the Pavia University data set. Each group contains two feature maps, including (Left) residual feature $\mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l)$ and (Right) output feature map $\boldsymbol{\phi}_{l+1}$. We randomly demonstrate 20 out of 64 learned feature map groups, revealing different structures that are activated by various convolutional filters.

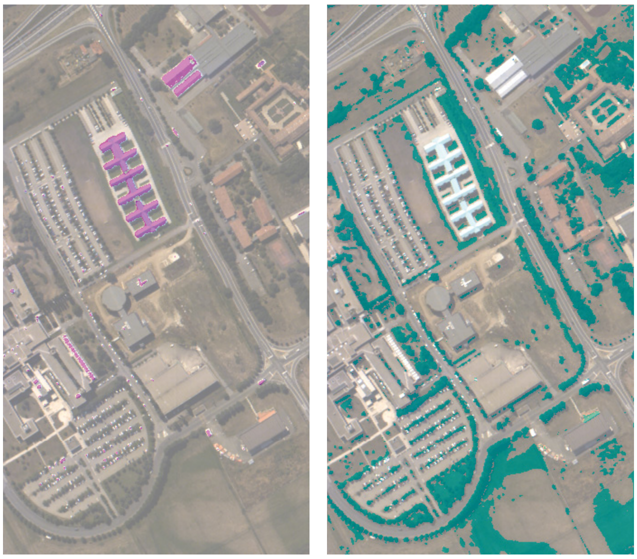
network once training is complete. Each group in Fig. 10 contains two feature maps, i.e., the residual feature $\mathcal{F}(\boldsymbol{\phi}_l; \boldsymbol{\Theta}_l)$ [Fig. 10 (left)] and the output feature $\boldsymbol{\phi}_{l+1}$ [Fig. 10 (right)] of the residual block. We randomly show 20 out of 64 learned feature map groups, revealing the different structures that are activated by various convolutional filters. For instance,



(a)

(b)

Fig. 11. (a) Eight out of 128 output feature maps of the second residual block. (b) Twelve out of 256 output feature maps of the third residual block.



(a)

(b)

Fig. 12. Object detection maps of selective convolutional filters from the first residual block of the proposed residual Conv–Deconv network, in which some "neurons" own good description power for semantic visual patterns in the object level. For example, the feature maps activated by the convolutional filters #52 and #03 in the first residual block can be used to precisely capture (a) metal sheets and (b) vegetative covers, respectively. Specifically, we achieve detection by simply setting a global threshold, which is computed by minimizing the intraclass variance of the black and white pixels in the considered feature map [53].

in group #47, the visualization of output feature map reveals that this particular feature focuses on the spectrum of metal sheets in the scene, while the output feature map in group #52 inhibits the expression of the same class. And, as shown in group #37, the residual feature tends to activate the shadow areas in the feature map. Since these feature maps are produced by the corresponding convolutional filters, it is believed that the convolutional filters learned by our residual Conv–Deconv network are capable of extracting some specific spectral–spatial patterns from different perspectives. We also show the output feature maps of the second and the third residual blocks in Fig. 11. It can be seen that the deeper the residual block is, the more abstract the learned feature maps will be naturally.

3) Object Detection: A very interesting thing arises when we analyze the learned feature maps. Although our residual Conv–Deconv network has not been explicitly designed for the task of object detection, we have observed strong evidence of object detection for the hyperspectral image provided by the network at the test stage. In particular, we found that target objects can be localized by the activated or suppressed pixels in some specific learned feature maps of the first residual block. For example, we can determine the objects consisted of metal sheets in the Pavia University data set through finding the hyperspectral pixels that are suppressed by the convolutional filter #52. Similarly, the vegetation covers, including meadows and trees, are able to be identified in the scene by searching the nonactivated pixels in the output feature map #03. To qualitatively assess the object detection results acquired by the proposed approach, examples of such object detection maps are given in Fig. 12. This visualization clearly demonstrates that some "neurons" in the first residual block of the proposed residual Conv–Deconv network know the locations of the target objects within the hyperspectral image and own good description power for semantic visual patterns in the object level. Addressing the detection task seems within reach. Moreover, it is worth noting that compared with the conventional supervised object detectors that need a number of labeled ground truth data, object detection achieved by this method is free and totally unsupervised. Also, as shown in Fig. 12, the quality of such object detection maps is quite good. These maps are with very good edge details, and even very small objects (e.g., cars on the road in the Pavia University scene) can be detected. In a nutshell, our study has shown that the convolutional filters in the proposed residual Conv–Deconv network for the task of unsupervised spectral–spatial feature learning possess strong selectiveness on patterns corresponding to object categories. Particularly, the feature maps obtained by some specific "neurons" at the first residual block of the network record the spectral–spatial representation of visual pattern of a specific object.

D. Fine-Tuned Network for Hyperspectral Image Classification

To further investigate the spectral-spatial features learned by the residual Conv–Deconv network, we evaluated the

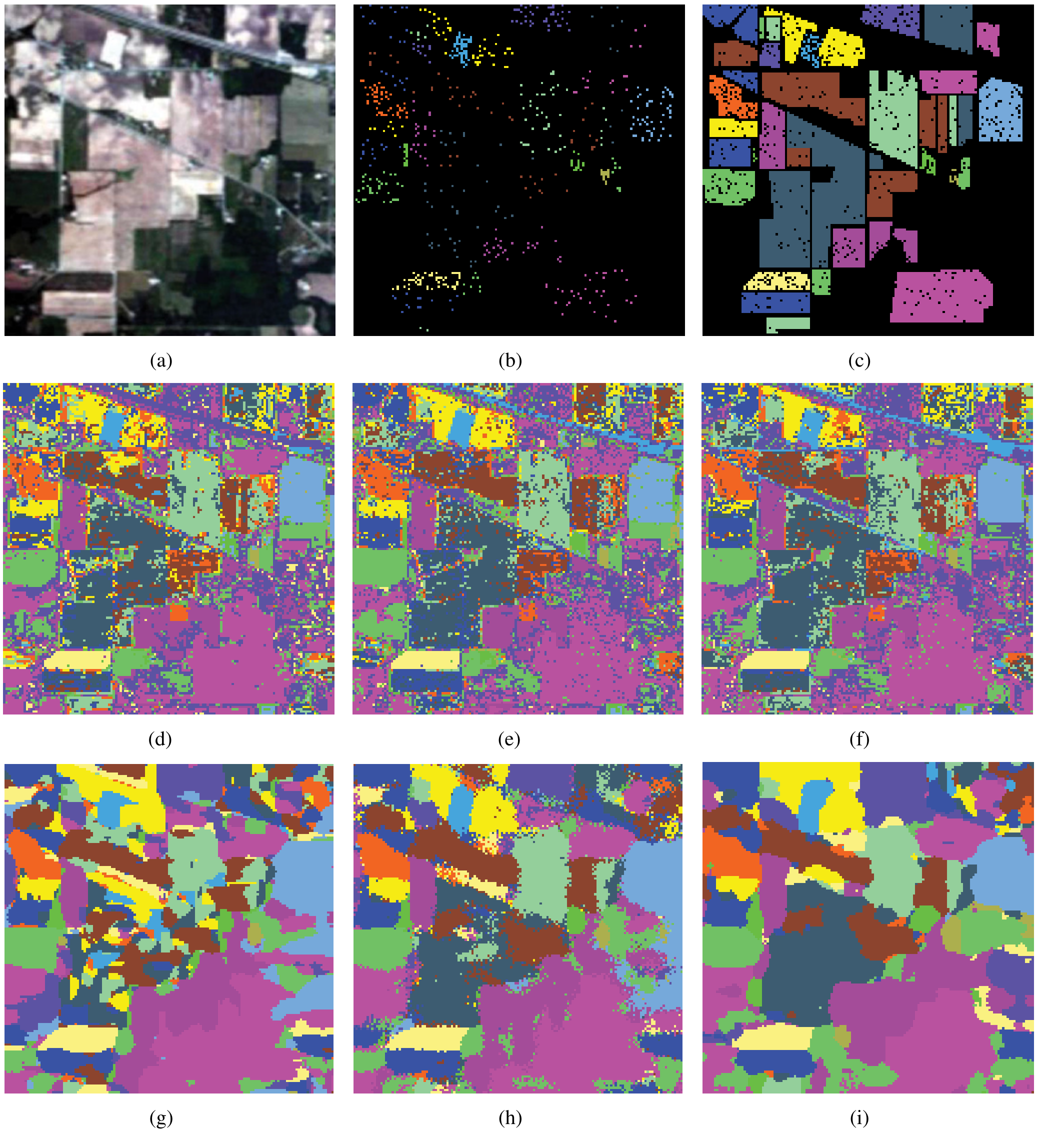


Fig. 13. Classification results obtained by different methods for the Indian Pines scene. (a) True-color composite (bands R: 26, G: 14, B: 8). (b) Training samples. (c) Test samples. (d) RF-200. (e) SVM-RBF. (f) 1-D CNN. (g) 2-D CNN. (h) SICNN. (i) Fine-tuned residual Conv–Deconv network.

performance of the fine-tuned network for the hyperspectral data classification task and provided a comparison with the state-of-the-art approaches.

The classification maps of the Indian Pines data set obtained by the widely used classifiers (e.g., random forest and SVM), supervised CNNs, and our method are shown in Fig. 13, and the corresponding accuracy indexes are presented in Table III. Analysis of the classification accuracy indexes indicates that the SVM with RBF kernel (SVM-RBF) outperforms the random forest classifier, mainly because the kernel SVM generally deals with nonlinear inputs more effectively than the random forest model. The proposed fine-tuned residual Conv–Deconv network achieves better scores for OA and kappa coefficient compared with all other methods. In comparison with SVM-RBF, 1-D CNN, and 2-D CNN, the proposed network increases the OA by 12.98%, 13.36%, and 15.97%, respectively. In addition, the numbers of test samples for different classes of Indian Pines are considerably imbalanced. Hence, the consideration of the OA alone cannot precisely evaluate the usefulness of the classifier, since small classes are commonly ignored. In this case, AA and kappa coefficient can be used to evaluate the performance of different classification models more accurately. Strong difference between the OA and AA or kappa coefficient may means that some classes are incorrectly classified with a high proportion. With respect to these two measures, compared with SVM-RBF, 1-D CNN, and 2-D CNN, the improvements in AA achieved by the proposed network are 9.89%, 12.20%, and 7.58%, respectively, and the increments of kappa coefficient obtained by the fine-tuned residual Conv–Deconv Net are 0.1454, 0.1533, and 0.1406, respectively. Note that the OA and kappa coefficient of 2-D CNN are significantly lower than those of other approaches, as directly training such 2-D network generally suffers from a small and imbalanced data set, while the

TABLE III
ACCURACY COMPARISON FOR THE INDIAN PINES DATA SET. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	Class Name	RF-200	SVM-RBF	1D CNN	2D CNN	SICNN	Res. Conv-Deconv Net
1	Alfalfa	55.71	60.77	56.79	66.98	**79.84**	74.86
2	Corn-notill	58.29	77.68	52.17	80.87	92.47	**95.28**
3	Corn-min	80.98	79.35	85.33	95.65	99.46	**100**
4	Corn	84.79	91.05	87.92	91.95	93.29	**95.08**
5	Grass-pasture	79.77	84.36	85.22	86.94	92.68	**96.56**
6	Grass-trees	95.90	92.03	97.49	97.95	96.58	**99.09**
7	Grass-pasture-mowed	76.58	69.61	74.62	67.86	**86.82**	84.42
8	Hay-windrowed	60.17	59.31	67.99	34.57	69.52	**74.57**
9	Oats	63.12	79.61	58.87	80.85	**83.69**	80.14
10	Soybean-notill	95.68	97.53	98.77	**100**	**100**	**100**
11	Soybean-mintill	88.75	85.21	87.62	88.18	**96.70**	95.74
12	Soybean-clean	53.33	63.64	72.42	91.52	**96.97**	96.06
13	Wheat	97.78	**100**	93.33	**100**	**100**	**100**
14	Woods	56.41	87.18	71.79	71.79	**94.87**	84.62
15	Buildings-grass-trees	81.82	90.91	90.91	**100**	**100**	**100**
16	Stone-steel-towers	**100**	**100**	**100**	**100**	**100**	**100**
OA	-	69.92	72.78	72.40	69.79	85.13	**85.76**
AA	-	76.82	82.39	80.08	84.70	**92.68**	92.28
Kappa	-	0.6605	0.6931	0.6852	0.6979	0.8313	**0.8385**

TABLE IV
CLASSIFICATION ACCURACIES OF DIFFERENT TECHNIQUES IN PERCENTAGE FOR PAVIA UNIVERSITY. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	Class Name	RF-200	SVM-RBF	1D CNN	2D CNN	SICNN	Res. Conv-Deconv Net
1	Asphalt	80.94	84.84	83.73	69.25	**84.21**	78.99
2	Meadows	55.91	67.09	65.70	93.39	91.10	**97.16**
3	Gravel	53.26	**72.13**	67.03	63.13	64.36	61.46
4	Trees	**98.76**	95.72	94.03	94.39	95.53	95.76
5	Metal Sheets	99.11	99.48	99.41	**100**	97.70	97.77
6	Bare Soil	79.26	93.30	**96.30**	49.06	56.53	59.46
7	Bitumen	83.76	91.88	**93.83**	72.26	77.29	79.5
8	Bricks	91.06	92.56	93.56	94.32	95.57	**96.82**
9	Shadows	98.10	97.47	**99.79**	93.77	96.20	92.40
OA	-	71.66	79.88	79.28	82.66	85.25	**87.39**
AA	-	82.24	88.27	**88.15**	81.06	84.28	84.37
Kappa	-	0.6517	0.7487	0.7423	0.7688	0.8041	**0.8308**

proposed strategy, to a large extent, is capable of overcoming this shortcoming. Moreover, SICNN also performs well on the Indian Pines data set, since the specially designed mechanism can effectively solve the curse of dimensionality and the lack of available training samples. But, it is worth noting that our method for feature learning is unsupervised, while 1-D CNN, 2-DCNN, and SICNN are supervised networks. Taking this into account, the performance of our approach is competitive and satisfactory. The proposed approach achieves the best accuracies on most of classes of the Indian Pines data set. For instance, the accuracy of the grass-pasture category obtained by fine-tuned residual Conv–Deconv network reaches 96.56%, and the proposed network can achieve 100% on the corn-min class.

Fig. 14 shows the classification maps using the Pavia University data set; the comparison of accuracies between the random forest, SVM-RBF, supervised CNNs, and our approach can be found in Table IV. It can be seen that the proposed fine-tuned residual Conv–Deconv network outperforms the others in terms of OA and kappa coefficient. Misclassification in this data set lies in similar objects, such as Meadow-Trees. The proposed network achieves the best AA of 96.46% on Meadow-Trees. Similarly, the misclassification problem in the Indian pines data set is also improved. For example, the AA of Corn-notill, Corn-min, and Corn obtained by the fine-tuned residual Conv–Deconv network is 96.79%, which is higher than that of SVM-RBF (82.69%), 1-D CNN (75.14%), 2-D CNN (89.49%), and SICNN (95.07%). Furthermore, in Figs. 13 and 14, it is obvious that the spectral classification methods (random forest, SVM, and 1-D CNN) always result in noisy scatter points in the classification maps, while the spectral–spatial approaches (2-D CNN, SICNN, and fine-tuned residual Conv–Deconv network) address this problem by eliminating noisy scattered points of misclassification.

In addition to comparing the proposed approach with the traditional classifiers (random forest and SVM) and other deep networks, some mathematical morphology-based methods like the morphological profile (MP) [54] are also considered in comparison due to their capacity to extract spatial features. Fauvel *et al.* [55] summarized some frequently used spectral–spatial features. Benediktsson *et al.* [56] proposed an extended MP (EMP) using principal component analysis (PCA) for hyperspectral image classification. The EMP-PCA [56] is able to achieve the OA of 77.7%, AA of 82.5%, and kappa coefficient of 0.71 on the Pavia University

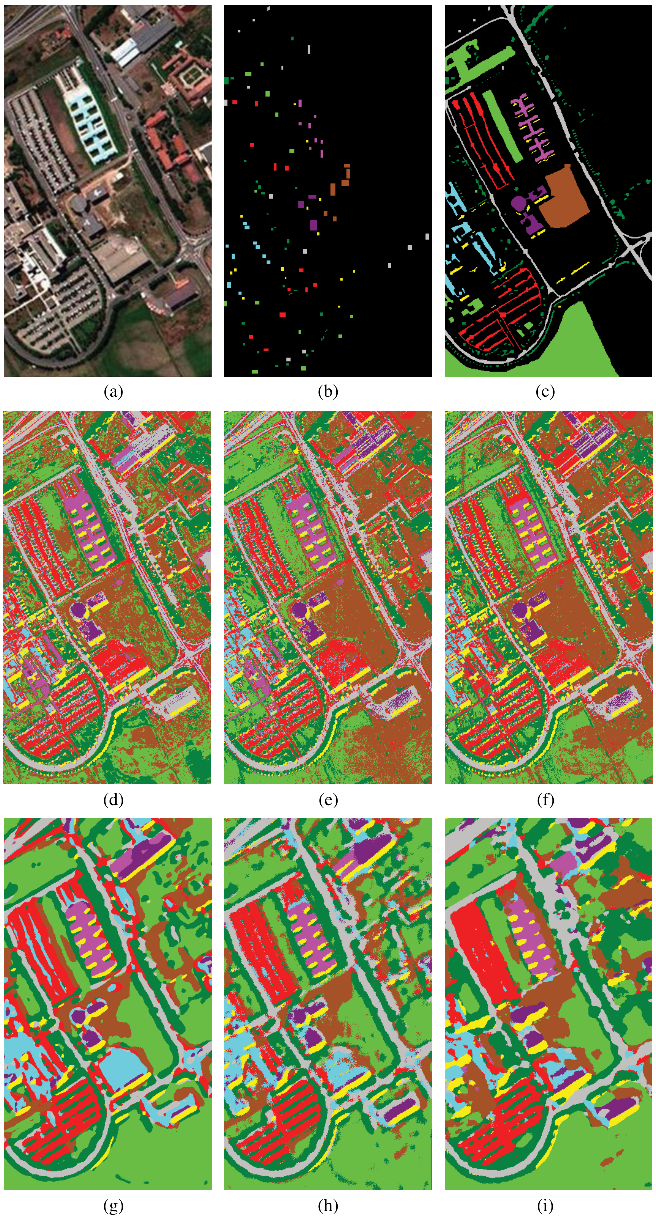


Fig. 14. Classification results obtained by different methods for the Pavia University scene. (a) Composite image of hyperspectral data. (b) Training data. (c) Ground truth reference. (d) RF-200. (e) SVM-RBF. (f) 1-D CNN. (g) 2-D CNN. (h) SICNN. (i) Fine-tuned residual Conv–Deconv network.

data set. Fauvel *et al.* [57] attempted to make use of kernel PCA to produce EMP, in which state-of-the-art performance on the Pavia University scene can be obtained with the OA of 96.3%, AA of 95.7%, and kappa coefficient of 0.95. For more mathematical morphology-based approaches, please refer to [55].

Table V gives information about the results of McNemar's test to evaluate the significance of the difference between the classification accuracies of the proposed network and the other investigated approaches. With reference to Table V, the improvements of OAs achieved by the proposed methods are statistically significant in comparison with the other studied methods. It is worth noting that the SICNN performs similarly to the proposed approach on the Indian Pines data set (the value is 1.747), as the SICNN exploits band selection before feeding the data into the CNN, which greatly reduces the total number of parameters of the network and thus improves the accuracy.

TABLE V

ASSESSMENT OF THE SIGNIFICANCE OF THE CLASSIFICATION ACCURACIES OF THE PROPOSED METHOD COMPARED WITH THE OTHER INVESTIGATED APPROACHES FOR BOTH THE INDIAN PINES AND PAVIA UNIVERSITY DATA SETS

Data set	RF-200	SVM-RBF	1D CNN	2D CNN	SICNN
Indian Pines	28.056	24.995	24.440	32.463	1.747
Pavia University	56.949	29.128	31.362	31.464	12.029

TABLE VI

STATISTICS OF TRAINING TIME (MINUTES)

Data set	Res. Conv-Deconv Net	Fine-tuned network
Indian Pines	20.3	3.1
Pavia University	34.8	6.9

E. Processing Time

For both training and testing steps of the residual Conv–Deconv network and the fine-tuned network, we have used an NVIDIA GTX Titan GPU. The other approaches, i.e., random forest, SVM-RBF, and 1-D CNN, are computed on a CPU with a personal computer equipped with an Intel Core I5 with 2.20 GHz. The training times of the residual Conv–Deconv network and the fine-tuned network are shown in Table VI. With the help of GPU, the training times of the proposed networks are acceptable.

V. CONCLUSION

In this paper, we proposed a novel end-to-end fully Conv–Deconv network architecture for unsupervised spectral–spatial feature extraction of hyperspectral images. In particular, the proposed network is composed of two parts, namely, the convolutional subnetwork and deconvolutional subnetwork. They are responsible for transforming an input 3-D hyperspectral patch to abstract feature representation and reproducing the initial input data from the encoded feature, respectively. Furthermore, residual learning and a new unpooling operation that can make use of max-pooling indexes are introduced to our network architecture in order to overcome the training problem caused by vanishing gradient. A very interesting observation can be found when we analyze the learned feature maps. Although the proposed network has not been explicitly designed for the task of object detection, we have observed that target object can be localized by the activated or suppressed pixels in some specific learned feature maps of the first residual block, which makes it possible to achieve the unsupervised object detection in hyperspectral images. Experimental results also demonstrate that the features learned by the proposed unsupervised network can be used for the hyperspectral image classification task, and the obtained classification results are competitive compared with the other supervised approaches.

In the future, further experiments and studies will be conducted to fully understand the "block box" of the proposed fully Conv–Deconv network with residual learning, providing more accurate analysis for remote sensing applications such as unsupervised object detection with the help of learned feature maps. In addition, the input to the proposed Conv–Deconv network is the raw hyperspectral data, and a possible future work is to explore the capability of the proposed approach using APs and extinction profiles that extract spatial information in a robust and adaptive way.

Acknowledgment

The authors would like to thank Prof. Paolo Gamba from the University of Pavia, Italy, for providing the ROSIS data and corresponding reference information and Prof. Landgrebe from Purdue University for providing the Indian Pines data set.

References

[1] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Boston, MA, USA: Artech House, 2015.

[2] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.

[3] Y. Gu, T. Liu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3235–3247, Jun. 2016.

[4] J. Li, M. Khodadadzadeh, A. Plaza, X. Jia, and J. M. Bioucas-Dias, "A discontinuity preserving relaxation scheme for spectral–spatial hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 625–639, Feb. 2016.

[5] P. Ghamisi, J. A. Benediktsson, and J. R. Sveinsson, "Automatic spectral–spatial classification framework based on attribute profiles and supervised feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5771–5782, Sep. 2014.

[6] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2858–2874, May 2014.

[7] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, p. 506, 2016.

[8] B. Demir, F. Bovolo, and L. Bruzzone, "Updating land-cover maps by classification of image time series: A novel change-detection-driven transfer learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 300–312, Jan. 2013.

[9] J. Meola, M. T. Eismann, R. L. Moses, and J. N. Ash, "Application of model-based change detection to airborne VNIR/SWIR hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3693–3706, Oct. 2012.

[10] L. G. Olmanson, P. L. Brezonik, and M. E. Bauer, "Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: The Mississippi River and its tributaries in Minnesota," *Remote Sens. Environ.*, vol. 130, pp. 254–265, Mar. 2013.

[11] M. S. Moran, Y. Inoue, and E. M. Barnes, "Opportunities and limitations for image-based remote sensing in precision crop management," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 319–346, Sep. 1997.

[12] S. Delalieux, B. Somers, B. Haest, T. Spanhove, J. V. Borre, and C. A. Mücher, "Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers," *Remote Sens. Environ.*, vol. 126, pp. 222–231, Nov. 2012.

[13] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[14] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[15] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[16] J. A. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2000, pp. 813–815.

[17] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.

[18] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[19] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.

[20] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[21] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[22] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.

[23] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[24] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, Feb. 2015.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, May 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[28] J. Hu, L. Mou, A. Schmitt, and X. X. Zhu, "FusioNet: A two-stream convolutional neural network for urban scene classification using PolSAR and hyperspectral data," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, 2017, pp. 1–4.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[32] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[33] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[34] L. Mou and X. X. Zhu, "Spatiotemporal scene interpretation of space videos via deep neural network and tracklet analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1823–1826.

[35] L. Mou *et al.*, "Multitemporal very high resolution from space: Outcome of the 2016 IEEE GRSS Data Fusion Contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3435–3447, Aug. 2017.

[36] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[37] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4959–4962.

[38] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[39] E. Aptoula, M. C. Ozdemir, and B. Yanikoglu, "Deep learning with attribute profiles for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1970–1974, Dec. 2016.

[40] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[42] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[43] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[44] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "Learning to generate chairs, tables and cars with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1538–1546.

[45] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[46] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.

[48] R. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *Proc. IEEE Int. Conf. Mach. Learn. (ICML)*, Jul. 2015.

[49] R. Goroshin, M. F. Mathieu, and Y. LeCun, "Learning to linearize under uncertainty," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1234–1242.

[50] G. M. Foody, "Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy," *Photogramm. Eng. Remote Sens.*, vol. 70, no. 5, pp. 627–633, 2004.

[51] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jan. 2015, Art. no. 258619.

[52] D. P. Kingma and J. Ba. (2015). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

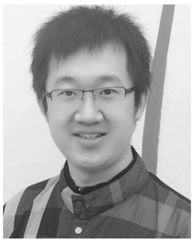
[53] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[54] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, Feb. 2001.

[55] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[56] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[57] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas," *EURASIP J. Adv. Signal Process.*, vol. 2009, p. 783194, Dec. 2009.



Lichao Mou (S'16) received the bachelor's degree in automation from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2012, and the master's degree in signal and information processing from the University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the German Aerospace Center, Wessling, Germany, and the Technical University of Munich, Munich, Germany.

In 2015, he joined the Computer Vision Group, University of Freiburg, Freiburg im Breisgau, Germany. His research interests include remote sensing, computer vision, and machine learning, especially remote sensing video analysis and deep networks with their applications in remote sensing.

Mr. Mou won the first place prize in the 2016 IEEE GRSS Data Fusion Contest.

Pedram Ghamisi (S'12–M'15) received the B.Sc. degree in civil (survey) engineering from the Tehran South Campus of Azad University, Tehran, Iran, the M.Sc. (First Class Hons.) degree in remote sensing from the K. N. Toosi University of Technology, Tehran, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2015.

In 2013 and 2014, he joined the School of Geography, Planning and Environmental Management, University of Queensland, Brisbane, QLD, Australia. He was a Post-Doctoral Research Fellow with the University of Iceland. He has been a Post-Doctoral Research Fellow with the Technical University of Munich, Munich, Germany, and Heidelberg University, Heidelberg, Germany, since 2015. He has also been a Researcher with the German Aerospace Center, Remote Sensing Technology Institute, Wessling, Germany, since 2015. His research interests include remote sensing and image analysis, with a special focus on the spectral and spatial techniques for hyperspectral image classification, multisensor data fusion, machine learning, and deep learning.

Dr. Ghamisi received the Best Researcher Award for M.Sc. students from the K. N. Toosi University of Technology. In 2013, he presented at the IEEE International Geoscience and Remote Sensing Symposium, Melbourne, VIC, Australia, and was awarded the IEEE Mikio Takagi Prize for winning the conference Student Paper Competition against almost 70 people. In 2016, he was selected as a Talented International Researcher by the Iran's National Elites Foundation. In 2017, he won the Data Fusion Contest 2017 organized by the Image Analysis and Data Fusion Technical Committee of the Geoscience and Remote Sensing Society. His model was the most accurate among more than 800 submissions. He received the prestigious Alexander von Humboldt Fellowship in 2015.

Xiao Xiang Zhu (S'10–M'12–SM'14) received the bachelor's degree in space engineering from the National University of Defense Technology, Changsha, China, in 2006, and the M.Sc., Dr.-Ing., and "Habilitation" degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She was a Guest Scientist or a Visiting Professor with the Italian National Research Council, Naples, Italy, in 2009; Fudan University, Shanghai, China, in 2014; the University of Tokyo, Tokyo, Japan, in 2015; and the University of California, Los Angeles, CA, USA, in 2016. She has been the Professor of signal processing in earth observation with TUM since 2015; the Head of the Team Signal Analysis, German Aerospace Center (DLR), Remote Sensing Technology Institute, since 2011; and the Head of the Helmholtz Young Investigator Group "SiPEO," DLR, and TUM, since 2013. Her research interests include advanced InSAR techniques such as high-dimensional tomographic SAR imaging and SqueeSAR; computer vision in remote sensing including object reconstruction and multidimensional data visualization; and big data analysis in remote sensing and modern signal processing, including innovative algorithms such as sparse reconstruction, nonlocal means filter, robust estimation, and deep learning, with applications in the field of remote sensing such as multi/hyperspectral image analysis.

Dr. Zhu is a member of young academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities and the German National Academy of Sciences Leopoldina and the Bavarian Academy of Sciences and Humanities. He is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

C Mou L., Bruzzone L., Zhu X., 2019. Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery, IEEE Transactions on Geoscience and Remote Sensing, 57(2), 924-935.

Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery

Lichao Mou, *Student Member, IEEE*, Lorenzo Bruzzone, *Fellow, IEEE*,
and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*— Change detection is one of the central problems in earth observation and was extensively investigated over recent decades. In this paper, we propose a novel recurrent convolutional neural network (ReCNN) architecture, which is trained to learn a joint spectral–spatial–temporal feature representation in a unified framework for change detection in multispectral images. To this end, we bring together a convolutional neural network and a recurrent neural network into one end-to-end network. The former is able to generate rich spectral-spatial feature representations, while the latter effectively analyzes temporal dependence in bitemporal images. In comparison with previous approaches to change detection, the proposed network architecture possesses three distinctive properties: 1) it is end-to-end trainable, in contrast to most existing methods whose components are separately trained or computed; 2) it naturally harnesses spatial information that has been proven to be beneficial to change detection task; and 3) it is capable of adaptively learning the temporal dependence between multitemporal images, unlike most of the algorithms that use fairly simple operation like image differencing or stacking. As far as we know, this is the first time that a recurrent convolutional network architecture has been proposed for multitemporal remote sensing image analysis. The proposed network is validated on real multispectral data sets. Both visual and quantitative analyses of the experimental results demonstrate competitive performance in the proposed mode.**

***Index Terms*— Change detection, long short-term memory (LSTM), multitemporal image analysis, recurrent convolutional neural network (ReCNN).**

I. Introduction

WITH the development of remote sensing technology, every day, massive amounts of remotely sensed data are produced from a rich number of spaceborne and airborne sensors; e.g., the Landsat 8 satellite is capable of imaging the entire Earth every 16 days in an 8-day offset from Landsat 7, and every 10 days, the Sentinel-2 mission can provide a global coverage of Earth's land surface. For the Sentinel-2 mission alone, to date about 3.4 PB of data have been acquired. Triggered by these exciting existing and future observation capabilities, methodological research on the multitemporal data analysis is of great importance [1], [2]. Change detection is very crucial in the field of multitemporal image analysis, as it is able to identify land use or land cover differences in the same geographical area across a period of time and can be used in a large number of applications, to name a few, urban expansion, disaster assessment, resource management, and monitoring dynamics of land use [3]–[5].

In the literature, many methods have been proposed to better identify land cover changes [1]. Among them, a widely used model is based on image algebra approaches. A classic one is change vector analysis (CVA) proposed by Malila [6]. CVA is designed to analyze possible multiple changes in pairs of multispectral pixels of bitemporal images. Bovolo and Bruzzone [7] propose a formal definition and a theoretical study of CVA in the polar domain. Later some extensions of the CVA model have been proposed, e.g., compressed CVA (C^2VA) [8]. CVA is used together with unsupervised threshold selection techniques based on different possible models of the data distribution. For example, the Rayleigh-Rice mixture density model [9] has been recently used in the framework of the expectation–maximization algorithm.

In addition, some image transformation-based models have been proposed in change detection to improve detection performance. These approaches mainly aim at learning a new, transformed feature representation from the original spectral domain, in order to suppress unchanged regions and highlight the presence of changes in the new feature space. For example, principal component analysis (PCA), Gram–Schmidt transformation, multivariate alteration detection (MAD), slow feature analysis (SFA), sparse learning, and deep belief network (DBN) use transformation algorithms in change detection methods [10]–[15]. PCA is one of the best-known subspace learning algorithms and can be used on both difference images and stacked images [10], [16]. The goal of Gram–Schmidt transformation is to reduce data correlation. MAD makes an attempt at maximizing the variance

Manuscript received March 6, 2018; revised June 7, 2018; accepted July 14, 2018. Date of publication November 20, 2018; date of current version January 21, 2019. This work was supported in part by the China Scholarship Council and the European Research Council through the European Unions Horizon 2020 Research and Innovation Programme under Grant ERC-2016-StG-714087 and in part by the Helmholtz Association through the Framework of the Young Investigators Group "SiPEO" under Grant VH-NG-1018. *(Corresponding author: Xiao Xiang Zhu.)*

L. Mou and X. X. Zhu are with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Weßling, Germany, and also with Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: lichao.mou@dlr.de; xiao.zhu@dlr.de).

L. Bruzzone is with the Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy (e-mail: lorenzo.bruzzone@unitn.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2863224

of independently transformed variables [12] and is invariant to linear scaling of the input data. SFA [13] is able to extract the most temporally invariant component from multitemporal images to transform data into a new feature space and, in this space, differences in unchanged pixels are suppressed so that changed regions can be better separated. Erturk *et al.* [14] apply sparse learning on stacked multitemporal images and expect that resulting sparse solutions do not vary greatly between the multitemporal data. Gong *et al.* [15] learn feature representations of two images with DBNs. Feature vectors issued from the two image acquisitions are stacked and used to learn a representation, where changes stand out more clearly. Using such feature representation, changes are more easily detected by image differencing.

Another important branch of change detection methods is based on classification approaches. For example, Bruzzone and Serpico [17] propose a supervised nonparametric model, based on the compound classification rule for minimum error, to detect land cover transitions between two remote sensing images acquired at different times. The main idea of this approach is to consider the temporal correlation between images in the classification without requiring complex training data. Bruzzone *et al.* [18] use the Bayes rule for minimum error in the compound classification framework for detecting land cover transitions between pairs multisource images gathered at two different dates. Bruzzone and Cossu [19] propose a multiclassifier architecture, which is composed of an ensemble of partially unsupervised classifiers, to detect changes or update land cover maps. Later, Bruzzone *et al.* [20] develop an effective system that employs an ensemble of nonparametric multitemporal classifiers to address the problem of detecting land cover transitions in multitemporal images. All these techniques consider different tradeoffs between modeling the temporal correlation in the training of the system and requiring complex training data.

One crucial issue in change detection is modeling the temporal correlation between bitemporal images. Various atmospheric scattering conditions, complicated light scattering mechanisms, and intraclass variability lead to change detection is inherently nonlinear. Thus sophisticated, task-driven, learning-based methods are desirable.

Deep neural networks have recently been shown to be very successful in a variety of computer vision and remote sensing tasks [21]. They can also provide the opportunity for change detection, where one would like to extract joint spectral-temporal features from a bitemporal image sequence in an end-to-end manner. In this respect, as an important branch of deep learning family, a recurrent neural network (RNN) is a natural candidate to tackle the temporal connection between multitemporal sequence data in change detection tasks. Recently, Lyu *et al.* [22] make use of an end-to-end RNN to solve the multispectral /hyperspectral image change detection task, since RNN is well known to be good at processing sequential data. In their framework, a long short-term memory (LSTM)-based RNN is employed to learn a joint spectral-temporal feature representation from a bitemporal image sequence. In addition, we also show the versatility of their network by applying it to detect multiclass changes and pointing out a good transferability for change detection in an “unseen” scene without fine-tuning. Russwurm and Körner [23] follow a similar idea, where an RNN based on LSTM units is used to extract dynamic spectral–temporal features but, in contrast to the change detection scenario, their goal is to address the land cover classification of the multitemporal image sequence.

In this paper, we would like to learn joint spectral–spatial–temporal features using an end-to-end network for change detection, which is named as a recurrent convolutional neural network (ReCNN), since it combines convolutional neural network (CNN) and RNN. Although both CNN [24]–[36] and RNN [22], [23], [37]–[39] are well-established techniques for remote sensing applications, to the best of our knowledge, we are the first to combine them for multitemporal data analysis in the remote sensing community. Note that integrating CNN and RNN in an end-to-end manner has also been explored in hyperspectral image classification [40], where the network is only used for extracting spectral information to build a spectral classifier for the classification purpose. In our work, the CNN part transforms the input, a pair of 3-D multispectral patches, to an abstract spectral-spatial feature representation, whereas the RNN part is not only employed for modeling temporal dependence, but is also used for predicting the final label (i.e., changed, unchanged, or change type). In other words, the features from the proposed ReCNN encapsulate information related to spectral, spatial, and temporal components in bitemporal images, making them useful for a holistic change detection task. For multitemporal image analysis, the proposed ReCNN contributes to the literature in three major aspects.

1) It is able to extract a spectral-spatial-temporal feature representation of multitemporal data through learning with a structured deep architecture.
2) It has the same property of 2-D CNN used for multispectral /hyperspectral data classification on learning informative spectral–spatial feature representations directly from multispectral data, requiring neither hand-crafted visual features nor preprocessing steps.
3) It has the same characteristic of RNN, being capable of modeling the temporal correlation between bitemporal images using a sophisticated and task-driven approach to the extraction of temporal features in an end-to-end architecture, and finally producing labels for the image sequence.

The remainder of this paper is organized as follows. After the introductory Section I detailing change detection, Section II is dedicated to the details of the proposed recurrent convolutional network. Section III then provides data set information, network setup, experimental results, and discussion. Finally, Section IV concludes this paper.

II. Methodology

A. Network Architecture

The architecture of the proposed ReCNN, as shown in Fig. 1, is made up of three components, including a convolutional subnetwork, a recurrent subnetwork, and fully connected layers, from bottom to top.

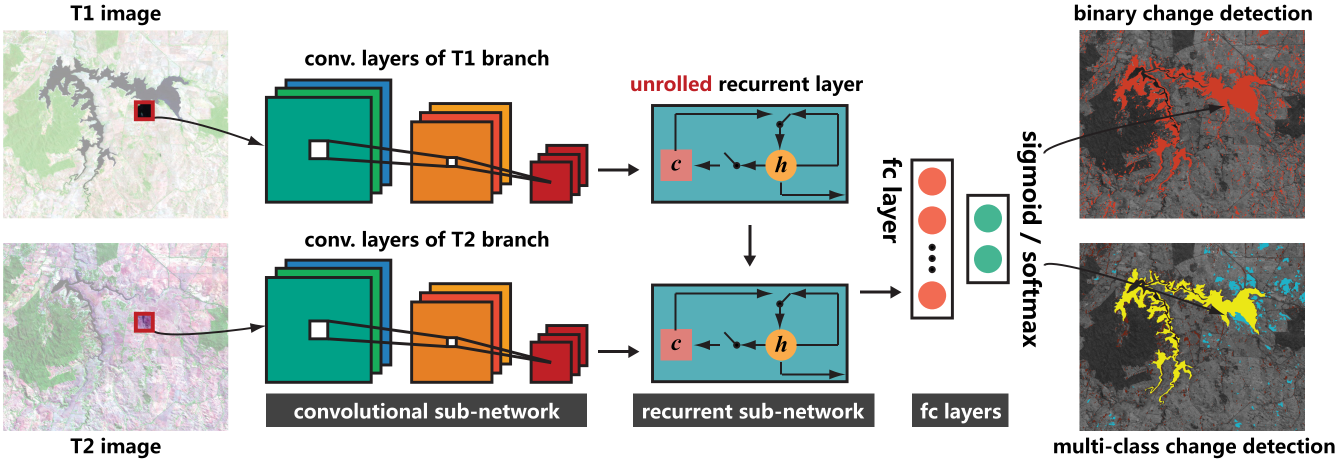


Fig. 1. Overview of the proposed ReCNN. At the bottom of our network, convolutional layers automatically extract feature maps from each input. On top of the convolutional subnetwork, a recurrent subnetwork takes the feature representations produced by convolutional layers as inputs to exploit the temporal dependence in the bitemporal images. To show how the single recurrent layer deals with bitemporal inputs, we show the unrolled form of the recurrent subnetwork. The third part is two fully connected layers widely used in classification problems. Although ReCNN is composed of different kinds of network architectures, i.e., CNN, RNN, and fully connected network, it can be trained end to end by backpropagation with one loss function, due to the differential property of all these components.

To acquire a joint spectral–spatial–temporal feature representation for change detection, at the bottom of our network, convolutional layers automatically extract feature maps from each input. On top of the convolutional subnetwork, a recurrent subnetwork takes the feature representations produced by convolutional layers as inputs to exploit the temporal dependence in the bitemporal images. The third part is two fully connected layers widely used in classification problems. Although ReCNN is composed of different kinds of network architectures (i.e., CNN, RNN, and fully connected network), it can be trained end to end by backpropagation with one loss function, due to the differential properties of all these components.

Let $\boldsymbol{X}^{T_1}$ and $\boldsymbol{X}^{T_2}$ represent a pair of multispectral images acquired over the same geographical area at two different times T_1 and T_2, respectively. Let $\boldsymbol{x}^{T_1}$ and $\boldsymbol{x}^{T_2}$ be two patches taken from the exact same location in two images. $\boldsymbol{y}$ is a label that indicates the category (i.e., changed, unchanged, or change type) that the pair of patches belongs to. The flowchart of the proposed ReCNN can be summarized as follows.

1) First, the 3-D multispectral patch $\boldsymbol{x}^{T_1}$ is fed into T_1 branch of the convolutional subnetwork, which transforms it to an abstract feature vector $\boldsymbol{f}^{T_1}$.
2) Then, the recurrent subnetwork receives $\boldsymbol{f}^{T_1}$ and calculates the hidden state information for the current input; it also restores that information in the meantime.
3) Subsequently, $\boldsymbol{x}^{T_2}$ is input to T_2 branch for extracting spectral–spatial feature $\boldsymbol{f}^{T_2}$, it is fed into the recurrent layer simultaneously with the state information of $\boldsymbol{f}^{T_1}$, and the activation at time T_2 is computed by a linear interpolation between existing value and the activation of the previous time T_1.
4) Finally, fully connected layers of the ReCNN predict the label of the input bitemporal multispectral patches by looping through the entire sequence.

The entire change detection map can be obtained by applying the network to all pixels in the image.

B. Spectral–Spatial Feature Extraction via the Convolutional Subnetwork

As we have mentioned, the spectral–spatial information is of great importance for change detection. Some of the previous widely used unsupervised image algebra-based and image transformation-based methods cannot totally capture task specialized features which may be discriminative for a specific change detection task. Features directly learned from data and driven by tasks are supposed to be better [21]. This advantage leads to our usage of a trainable feature generator.

Though trainable, early and fairly simple 1-D neural network models, such as DBN [15] and multilayer perceptron (MLP), suffer from huge amount of learnable parameters, since those architectures are totally equipped with fully connected layers, which is an undesirable case given that available annotated training samples for change detection are often very limited. Moreover, another disadvantage of such networks is that they treat the multispectral data as vectors, ignoring the 2-D property of imagery in the spatial domain.

CNNs, which are a significant branch of deep learning, have been attracting attention, due to the fact that they are capable of automatically discovering relevant contextual 2-D spatial features as well as spectral features for multispectral/hyperspectral data. In addition, a CNN makes use of local connections to deal with spatial dependencies via sharing weights, and thus can significantly reduce the number of parameters of the network in comparison with the conventional 1-D fully connected neural networks, e.g., DBN and MLP. Recently, CNNs used for hyperspectral image classification have proven their effectiveness in extracting useful spectral–spatial features [28], [41]. Triggered by this, adopting a CNN in our architecture is natural.

However, a direct use of CNNs commonly used in typical recognition tasks, e.g., AlexNet [42], VGG Nets [43], and GoogLeNet [44], is not possible in our task, as we believe that a simpler network architecture is more appropriate for our

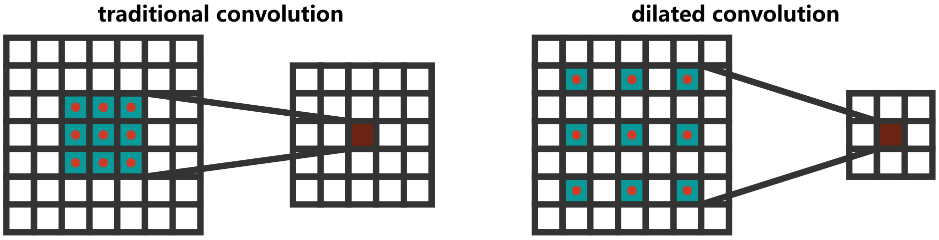


Fig. 2. Illustration of (Left) traditional convolution operation and (Right) two-dilated convolution. Traditional convolution corresponds to dilated convolution with dilation rate 1. Employing dilated convolution operation enlarges the network's field of view.

problem due to the following reasons. First, change detection aims to distinguish only several classes (two for binary change detection), which requires much less model complexity than general visual recognition problems in computer vision, such as ImageNet classification with 1000 categories. Second, since the spatial resolution of multispectral imagery is limited, it is desirable to make input size small, which reduces the depth of the network naturally. Third, a smaller network is obviously more efficient in change detection problems, where testing may be performed in a large-scale area. Finally, the above-mentioned networks are not suitable to be used on multispectral images with a large number of spectral channels.

The convolutional subnetwork receives a sequence of 5×5 multispectral patches as the input and has two separate, yet identical convolutional branches (i.e., T_1 branch and T_2 branch (cf. Fig. 1) which process $\boldsymbol{x}^{T_1}$ and $\boldsymbol{x}^{T_2}$ in parallel, respectively. The learned features are fed into the following recurrent subnetwork. Using this two-branch architecture, the convolutional RNN is constrained to first learn meaningful spectral–spatial representations of input patches, and to combine them on a higher level for modeling temporal dependence. More specifically, we make use of convolutional filters with a very small receptive field of 3×3, rather than using larger ones such as 5×5. Moreover, we do not adopt max-pooling after convolution or spatial padding for convolutional layers. The depth of the convolutional subnetwork is such that the output size of the last layer is 1×1.

Regarding convolution, we make use of dilated convolution to construct convolutional layers in the network because, for our task, it is able to offer a slightly better performance than a traditional convolution operation. The dilated convolution [45] was originally designed for the efficient computation of the undecimated wavelet transform in the "algorithme à trous" scheme [46]. This algorithm makes it possible to calculate responses of any layer at any desirable resolution and can be applied *post hoc*, once a network has been trained. Let $F : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be a discrete function. Let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and let $k : \Omega_r \rightarrow \mathbb{R}$ be a discrete filter of size $(2r+1)^2$. The traditional discrete convolution operation $*$ can be defined as follows:

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t). \tag{1}$$

This operation can be generalized. Let l be a dilation rate and let $*_l$ be defined as

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t). \tag{2}$$

We will refer to $*_l$ as a dilated convolution or an l-dilated convolution. Fig. 2 shows differences between the conventional convolution and the dilated convolution.

The usage of dilated convolution in our network allows us to exponentially enlarge the field of view with a linearly increasing number of parameters, providing a significant parameter reduction while increasing effective field of view. Note that a very recent study [47] found that large field of view actually plays an important role. This can be easily understood by an analogy that states the fact that humans usually confirm the category of a pixel by referring to its surrounding context region.

C. Modeling Temporal Dependence by the Recurrent Subnetwork

The impressive success of recent deep learning systems has been predominantly achieved by feedforward neural network architectures such as CNN. In such networks, we implicitly assume that all inputs are independent of each other. However, for tasks that involve processing time sequence (e.g., change detection), that is not a good assumption. RNNs are a kind of neural networks that extend the conventional feedforward neural networks with loops in connections. Unlike a feedforward network, an RNN is capable of dealing with dependent, sequential inputs by having a recurrent hidden state whose activation at each time step depends on that of the previous time. By doing so, the network can exhibit dynamic temporal behavior, which is in line with our purpose, i.e., modeling temporal dependence between the T_1 and T_2 data. To this end, three types of RNN architectures, namely, fully connected RNN, LSTM, and gated recurrent unit (GRU), are used to construct the recurrent subnetwork in our ReCNN.

1) Fully Connected RNN: Given feature vectors $\boldsymbol{f}^{T_1}$ and $\boldsymbol{f}^{T_2}$ learned from the convolutional subnetwork, a fully connected RNN updates its recurrent hidden state $\boldsymbol{h}_t$ by

$$\boldsymbol{h}_t = \begin{cases} 0 & \text{if } t = 0 \\ \varphi(\boldsymbol{h}_{t-1}, \boldsymbol{f}^{T_t}) & \text{otherwise} \end{cases} \tag{3}$$

where φ is a nonlinear activation function, such as a hyperbolic tangent function or logistic sigmoid function. The recurrent layer will output a sequence $\boldsymbol{h} = (\boldsymbol{h}_1, \boldsymbol{h}_2)$. For our task, we only need the last one as input to the fully connected layers for predicting label.

In the fully connected RNN model, the update of the recurrent hidden state in (3) is implemented as

$$\boldsymbol{h}_t = \varphi(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{f}^{T_t}) \tag{4}$$

where $\boldsymbol{U}$ and $\boldsymbol{W}$ are the coefficient matrices for the activation of recurrent hidden units at the previous time step and for the input at the present time, respectively.

Fully connected RNN is the concisest RNN model, and it can reflect the essence of RNNs, i.e., an RNN is capable of modeling a probability distribution over the next element of the sequence data, given its present state $\boldsymbol{h}_t$, by capturing a distribution over sequence data. Let $p(\boldsymbol{f}^{T_1}, \boldsymbol{f}^{T_2})$ be the sequence probability, which can be decomposed into

$$p(\boldsymbol{f}^{T_1}, \boldsymbol{f}^{T_2}) = p(\boldsymbol{f}^{T_1})p(\boldsymbol{f}^{T_2}|\boldsymbol{f}^{T_1}). \tag{5}$$

Then, the conditional probability distribution can be modeled with an RNN

$$p(\boldsymbol{f}^{T_2}|\boldsymbol{f}^{T_1}) = \varphi(\boldsymbol{h}_2) \tag{6}$$

where $\boldsymbol{h}_2$ is obtained from (3). More specifically, the RNN tries to model the conditional dependence between a patch at T_1 and its corresponding one at T_2 in the following manner:

$$\begin{aligned} p(\boldsymbol{f}^{T_2}|\boldsymbol{f}^{T_1}) &= \varphi(\boldsymbol{U}\boldsymbol{h}_1 + \boldsymbol{W}\boldsymbol{f}^{T_2}) \\ &= \varphi(\boldsymbol{U}\varphi(\boldsymbol{W}\boldsymbol{f}^{T_1}) + \boldsymbol{W}\boldsymbol{f}^{T_2}). \end{aligned} \tag{7}$$

In this way, a conditional probability distribution p, which is beneficial to our change detection tasks, can be modeled by optimizing $\boldsymbol{W}$ and $\boldsymbol{U}$ during task-guided network training. Our motivation in this paper is apparent here: bitemporal images act as true sequential data instead of a simple difference image or stacked image and, therefore, an RNN can be used to model the temporal dependence.

2) LSTM: LSTM is a special type of recurrent hidden unit and was initially proposed by Hochreiter and Schmidhuber [48]. Since then, a couple of minor modifications to the original version have been made. In this paper, we follow the implementation of LSTM as used in [49]. As shown in (3), recurrent hidden units in a fully connected RNN simply compute a weight sum of inputs and then apply a nonlinear function. In contrast, an LSTM-based recurrent layer maintains a series of memory cells $\boldsymbol{c}_t$ at time step t. The activation of LSTM units can be calculated by

$$\boldsymbol{h}_t = \boldsymbol{o}_t \tanh(\boldsymbol{c}_t) \tag{8}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function and $\boldsymbol{o}_t$ is the output gates that control the amount of memory content exposure. The output gates are updated by

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{oi}\boldsymbol{f}^{T_t} + \boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{oc}\boldsymbol{c}_t) \tag{9}$$

where the $\boldsymbol{W}$ terms represent coefficient matrices; for example, $\boldsymbol{W}_{oi}$ and $\boldsymbol{W}_{oc}$ are the input–output weight matrix and memory-output weight matrix, respectively.

The memory cells $\boldsymbol{c}_t$ are updated by partially discarding the present memory contents and adding new contents of the memory cells $\tilde{\boldsymbol{c}}_t$

$$\boldsymbol{c}_t = \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} \tag{10}$$

where $\odot$ is an elementwise multiplication. The new memory contents are

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_{ci}\boldsymbol{f}^{T_t} + \boldsymbol{W}_{ch}\boldsymbol{h}_{t-1}) \tag{11}$$

where $\boldsymbol{W}_{ci}$ is input-memory weight matrix and $\boldsymbol{W}_{ch}$ represents hidden-memory coefficient matrix.

The $\boldsymbol{i}_t$ and $\boldsymbol{f}_t$ are the input gates and forget gates, respectively. The former modulates the extent to which the new memory information is added to the memory cell, whereas the latter controls the degree to which contents of the existing memory cells are forgotten. Specifically, gates are computed as follows:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ii}\boldsymbol{f}^{T_t} + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ic}\boldsymbol{c}_{t-1}) \tag{12}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fi}\boldsymbol{f}^{T_t} + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{fc}\boldsymbol{c}_{t-1}). \tag{13}$$

3) GRU: Similar to LSTM, a GRU makes use of a linear sum between the existing state and the newly computed state. It, however, directly exposes whole state values at each time step, instead of controlling what part of the state information will be exposed.

The activation $\boldsymbol{h}_t$ of GRUs at time step t is a linear interpolation between the previous activation $\boldsymbol{h}_{t-1}$ and the candidate activation $\tilde{\boldsymbol{h}}_t$

$$\boldsymbol{h}_t = (\boldsymbol{1} - \boldsymbol{u}_t)\boldsymbol{h}_{t-1} + \boldsymbol{u}_t\tilde{\boldsymbol{h}}_t \tag{14}$$

where the update gates $\boldsymbol{u}_t$ determine how much GRUs update their activations or contents. Update gates can be computed by

$$\boldsymbol{u}_t = \sigma(\boldsymbol{W}_{ui}\boldsymbol{f}^{T_t} + \boldsymbol{W}_{uh}\boldsymbol{h}_{t-1}) \tag{15}$$

where $\boldsymbol{W}_{ui}$ and $\boldsymbol{W}_{uh}$ are the input-update coefficient matrix and hidden-update weight matrix, respectively.

The candidate activation $\tilde{\boldsymbol{h}}_t$ is computed similar to that of the fully connected RNN [cf. (3)] and as follows:

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{W}\boldsymbol{f}^{T_t}) \tag{16}$$

where $\boldsymbol{r}_t$ is the set of reset gates. When reset gates are totally OFF (i.e., $\boldsymbol{r}_t$ is $\boldsymbol{0}$), GRUs will completely forget the activation of the recurrent layer at previous time and only receive existing input. When open, reset gates will partially keep the information of the previously computed state. Reset gates are calculated similar to update gates

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_{ri}\boldsymbol{f}^{T_t} + \boldsymbol{W}_{rh}\boldsymbol{h}_{t-1}) \tag{17}$$

where $\boldsymbol{W}_{ri}$ is the input-reset weight matrix and $\boldsymbol{W}_{rh}$ represents the hidden-reset coefficient matrix.

Fig. 3 shows graphic models of fully connected RNN, LSTM, and GRU through time.

D. Network Training

The network training is based on the TensorFlow framework. We chose Nesterov Adam [50], [51] as the optimizer to train the network since, for this task, it shows much faster convergence than standard stochastic gradient descent with momentum [52] or Adam [53]. We fixed almost all of parameters of Nesterov Adam as recommended in [50]: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1\text{e}{-08}$, and a schedule decay of 0.004, making use of a fairly small learning rate of 2e−04. All network weights are initialized with a Glorot uniform initializer [54] that draws samples from a uniform distribution. We utilize sigmoid and softmax as activation functions of the

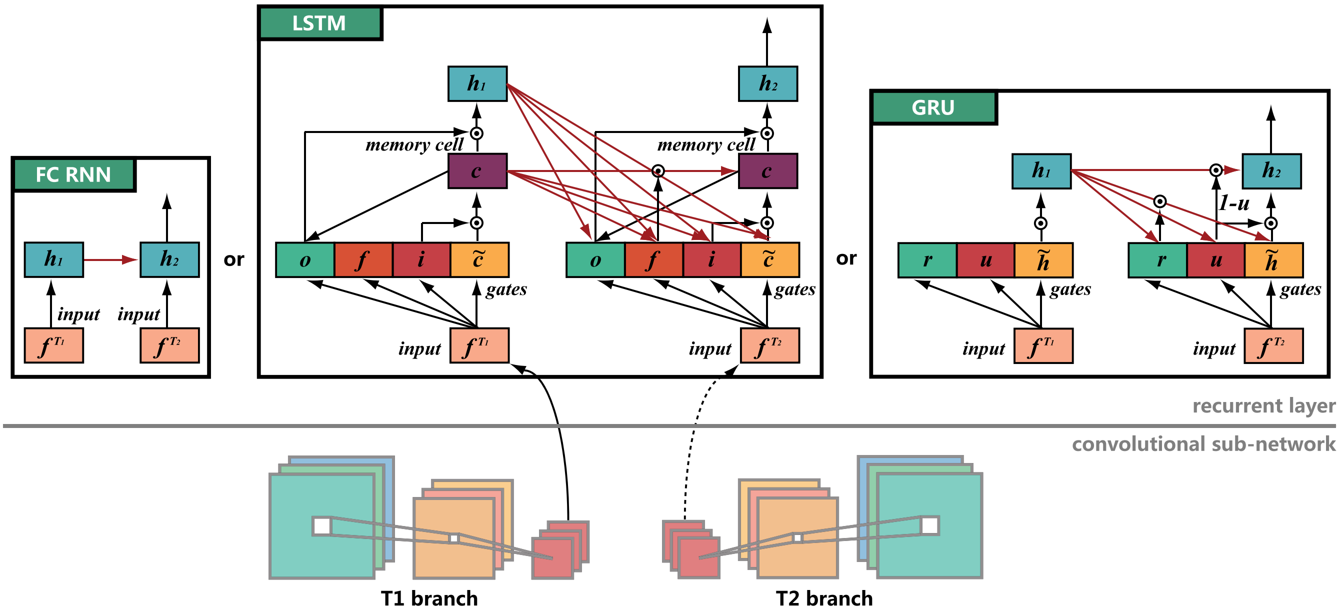
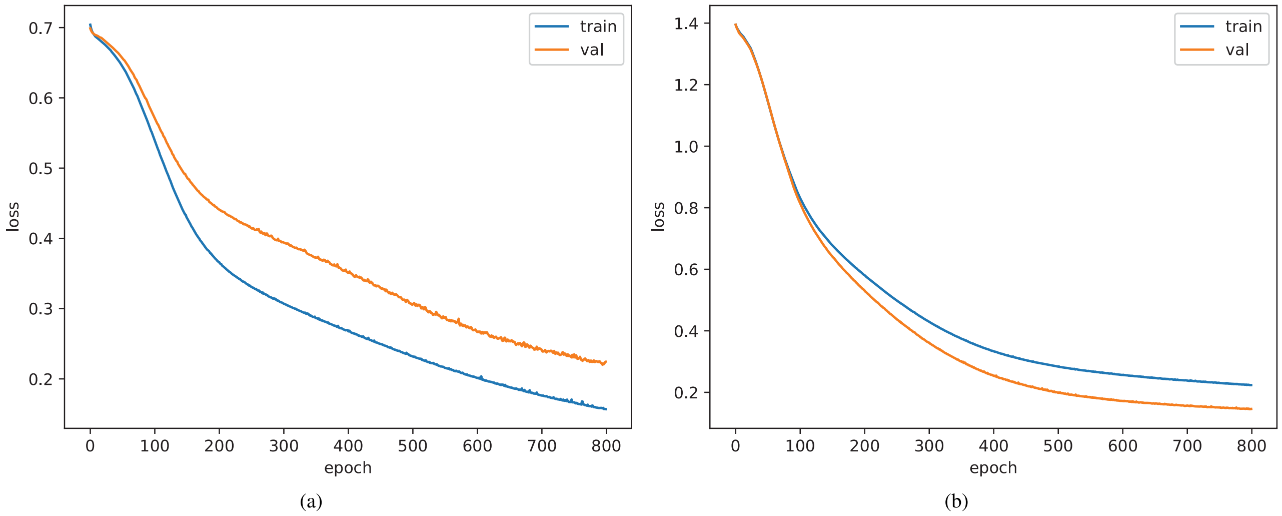


Fig. 3. Graphic models of fully connected RNN, LSTM, and GRU. In LSTM, $\boldsymbol{o}$, $\boldsymbol{f}$, $\boldsymbol{i}$, $\tilde{\boldsymbol{c}}$, and $\boldsymbol{c}$ are output gates, forget gates, input gates, new memory cell contents, and memory cells, respectively. In GRU, the reset and update gates are denoted by $\boldsymbol{r}$ and $\boldsymbol{u}$, respectively, and $\tilde{\boldsymbol{h}}$ and $\boldsymbol{h}$ are the candidate activation and final activation, respectively.



(a)

(b)

Fig. 4. Loss curves of ReCNN on the Taizhou data set for (a) binary change detection and (b) multiclass change detection tasks.

last fully connected layer for the binary and multiclass change detections, respectively. For the final loss, cross-entropy is chosen, which can be described as follows:

$$E = -\sum_{i} y_i \log \hat{y}_i \tag{18}$$

where $\hat{y}_i$ is the predicted probability value for class i. We use fairly small minibatches of 64 patch pairs. Moreover, we train the network for 800 epochs. There are no regularization techniques used in network training. We do not perform data augmentation before training the network. Finally, we train our network on a single NVIDIA GeForce GTX TITAN with 12 GB of GPU memory.

Fig. 4 shows loss curves of the proposed network during the training phase.

III. Experimental Results and Discussion

A. Data Description

The performance of the proposed network is evaluated on two data sets, which were acquired by the Landsat Enhanced Thematic Mapper Plus (ETM+) sensor with six bands and a spatial resolution of 30 m. Before feeding data into models, digital numbers of the original data were converted into absolute radiance (i.e., all of the data sets used in the experiments were normalized into a range of [0, 1]).

1) Taizhou Data: This data set consists of two images covering the city of Taizhou, China, in March 2000 and February 2003, with a WGS-84 projection and a coordinate range of 31°14′56N–31°27′39N, 120°02′24E–121°07′45E. These two images both consist of 400 × 400 pixels, and

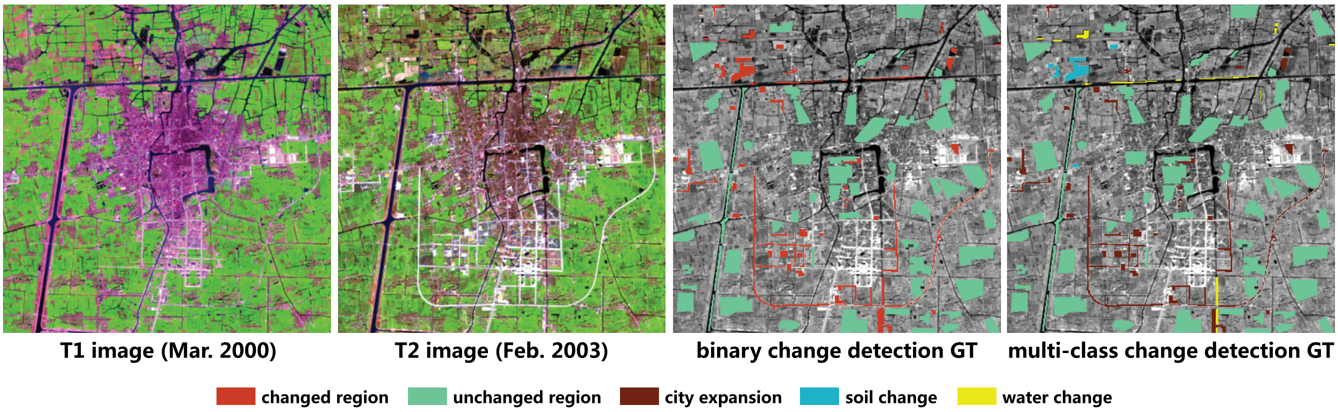


Fig. 5. True-color composites of the T_1 and T_2 images in the Taizhou data set as well as GTs.

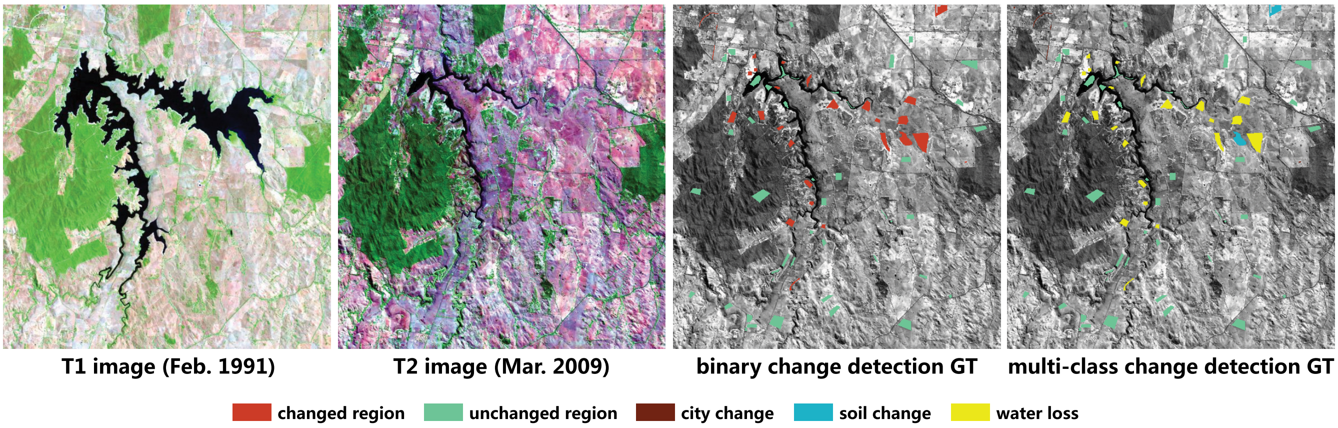


Fig. 6. Eppalock lake data set.

TABLE I
NUMBER OF TRAINING AND TEST SAMPLES IN THE TAIZHOU DATA SET

	Class Name	Training	Test
Binary	Changed region	500	4055
	Unchanged region	500	16961
	TOTAL	1000	21016
Multiple	Unchanged region	500	16961
	City expansion	500	2875
	Soil change	500	104
	Water change	500	75
	TOTAL	2000	20015

TABLE II
NUMBER OF TRAINING AND TEST SAMPLES IN THE EPPALOCK LAKE DATA SET

	Class Name	Training	Test
Binary	Changed region	500	3380
	Unchanged region	500	4515
	TOTAL	1000	7895
Multiple	Unchanged region	300	4715
	Water loss	300	2817
	Soil change	300	341
	City change	50	72
	TOTAL	950	7945

the changes between them mainly involve city expansion. The available manually annotated samples of this data set for multiclass change detection cover four classes of interest (cf. Fig. 5), i.e., unchanged area, city change/expansion (bare soils, grasslands, or cultivated fields to buildings, or roads), soil change (cultivated field to bare soil), and water change (nonwater regions to water regions). Table I provides information about different classes and their corresponding training and test samples.

2) Eppalock Lake: The second data set was acquired over the Eppalock lake, Victoria, Australia, in February 1991 and March 2009, with a WGS-84 projection and a coordinate range of 36°49′10S–37°00′52S, 144°27′52E–144°37′35E. Both images in this data set are 602×631 pixels. Similar to the Taizhou data, four multiclass change types are considered in the Eppalock lake scene, and they are unchanged region, city change (buildings or roads to bare soils, grasslands, or cultivated fields), water loss (water regions to bare soils), and soil change (vegetative covers or artificial buildings to bare soils). Fig. 6 shows tow true-color composite images and their corresponding reference samples. The number of training and test samples is displayed in Table II.

B. General Information

To evaluate the performance of different change detection algorithms, we utilize the following evaluation criteria.

TABLE III
ACCURACY COMPARISON OF BINARY CHANGE DETECTION ON THE TWO EXPERIMENTAL DATA SETS

	Taizhou City				Eppalock Lake			
	OA	Kappa	Unchanged	Changed	OA	Kappa	Unchanged	Changed
CVA [7]	83.82	0.3202	97.38	27.10	81.28	0.6353	69.24	97.37
PCA [10]	94.63	0.8181	**99.79**	74.51	74.68	0.5044	64.98	87.63
MAD [12]	94.62	0.8168	98.47	78.52	91.10	0.8138	99.14	80.36
IRMAD [55]	95.14	0.8313	99.35	77.53	91.27	0.8174	**99.49**	80.30
CNN [56]	96.03	0.8667	98.97	83.75	95.00	0.8975	97.34	91.89
RNN [22]	96.50	0.8884	97.58	91.96	95.21	0.9018	97.03	92.78
ReCNN-FC	98.35	0.9470	98.94	95.86	98.40	0.9674	98.56	98.20
ReCNN-GRU	98.67	0.9571	99.23	96.30	98.64	0.9723	99.22	97.87
ReCNN-LSTM	**98.73**	**0.9592**	99.20	**96.77**	**98.67**	**0.9728**	98.83	**98.46**

1) Overall accuracy (OA): This index shows the number of bitemporal pixels that are classified correctly, divided by the number of test samples.
2) Kappa coefficient: This metric is a statistical measurement of agreement between the final change detection map and the ground-truth (GT) map. It is the percentage agreement corrected by the level of agreement that could be expected due to change alone. In general, it is thought to be a more robust measure than a simple percent agreement computation, as k takes into account the agreement occurring by chance.

To validate the effectiveness of the proposed ReCNN model, it is compared with the most widely used change detection methods. These methods are summarized as follows.

1) CVA [7], which is an effective unsupervised approach for multispectral image change detection tasks.
2) PCA [10], which is simple in computation and can be applied to real-time applications.
3) MAD [12], which is a classical image transformation-based unsupervised algorithm for bitemporal multispectral image change detection.
4) Iteratively reweighted MAD (IRMAD) [55], which is an extension to MAD by introducing an iterative scheme.
5) Decision tree (DT), which is a nonparametric supervised learning method used for classification and regression. Its goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from data features.
6) Support vector machine (SVM), which works by mapping data to a kernel-included high-dimensional feature space seeking an optimal decision hyperplane that can best separate data samples, when data points are not linearly separable. Here, we use an SVM with radial basis function (RBF) kernel. The optimal hyperplane parameters C (parameter that controls the amount of penalty during the SVM optimization) and γ (spread of the RBF kernel) have been traced in the range of $C = 10^{-2}, 10^{-1}, \ldots, 10^{4}$ and $\gamma = 2^{-3}, 2^{-2}, \ldots, 2^{4}$ using fivefold cross validation.
7) CNN [56], a deep learning-based method, has proven successful in pattern recognition problems of hyperspectral imagery.

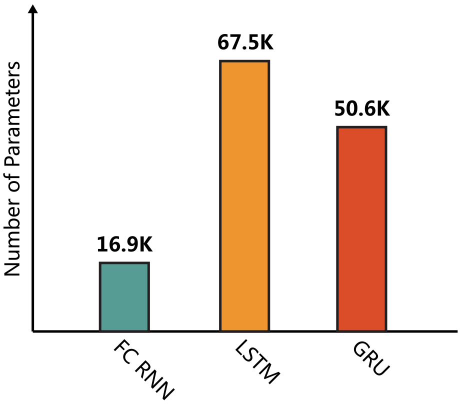


Fig. 7. Comparisons of different RNN architectures in terms of model size. Here, 128 recurrent units are used in each architecture.

8) RNN [22], a deep learning-based method, has recently shown promising performance in classification and change detection.
9) ReCNN-FC, which uses fully connected RNN as recurrent subnetwork in ReCNN model.
10) ReCNN-GRU, which uses GRU architecture in the recurrent subnetwork.
11) ReCNN-LSTM, which is the ReCNN model with LSTM as recurrent component.

Among these methods, CVA, PCA, MAD, IRMAD, and RNN are used in binary change detection experiments, and DT, SVM, and RNN are compared to the proposed network in multiclass change detection experiments. Moreover, k-means algorithm is used to automatically select threshold for unsupervised methods in the binary change detection task.

C. Analysis of Recurrent Subnetwork: Comparisons Between Fully Connected RNN, LSTM, and GRU

The most prominent trait shared between fully connected RNN, LSTM, and GRU is that there exists an additive loop of their update from T_1 to T_2, which is lacking in the conventional feedforward neural networks such as CNNs. In contrast, compared to the fully connected RNN like (4), both LSTM and GRU keep the current content and add the new content on top of it [cf. (10) and (14)]. These two RNN architectures, however, have a number of differences as well. LSTM makes use of three gates and a cell, namely, an input gate, forget gate, output gate, and memory cell, to control the exposure of memory content; whereas GRU only utilizes two gates to

TABLE IV
ACCURACY COMPARISON OF MULTICLASS CHANGE DETECTION ON THE TWO EXPERIMENTAL DATA SETS

		OA	Kappa	Unchanged	City change	Soil change	Water change/loss
Taizhou City	Decision Tree	85.19	0.5846	84.64	88.49	82.69	86.67
	SVM	93.90	0.7927	94.69	89.32	92.31	93.33
	CNN [56]	94.82	0.8155	96.56	85.11	88.46	82.67
	RNN [22]	95.48	0.8374	97.04	86.92	85.58	85.33
	ReCNN-FC	97.37	0.9039	97.95	94.12	**95.19**	92.00
	ReCNN-GRU	97.52	0.9097	98.05	94.54	**95.19**	96.00
	ReCNN-LSTM	**98.04**	**0.9279**	**98.36**	**96.31**	94.23	**97.33**
Eppalock Lake	Decision Tree	87.56	0.7811	81.31	41.67	89.15	99.01
	SVM	95.86	0.9228	94.46	72.22	97.65	98.58
	CNN [56]	95.49	0.9156	94.27	20.83	97.95	99.15
	RNN [22]	96.34	0.9392	95.55	41.67	96.48	99.04
	ReCNN-FC	98.45	0.9705	98.01	80.56	**100**	**99.47**
	ReCNN-GRU	98.49	0.9712	98.24	79.17	**100**	99.22
	ReCNN-LSTM	**98.70**	**0.9752**	**98.49**	**84.72**	**100**	99.25

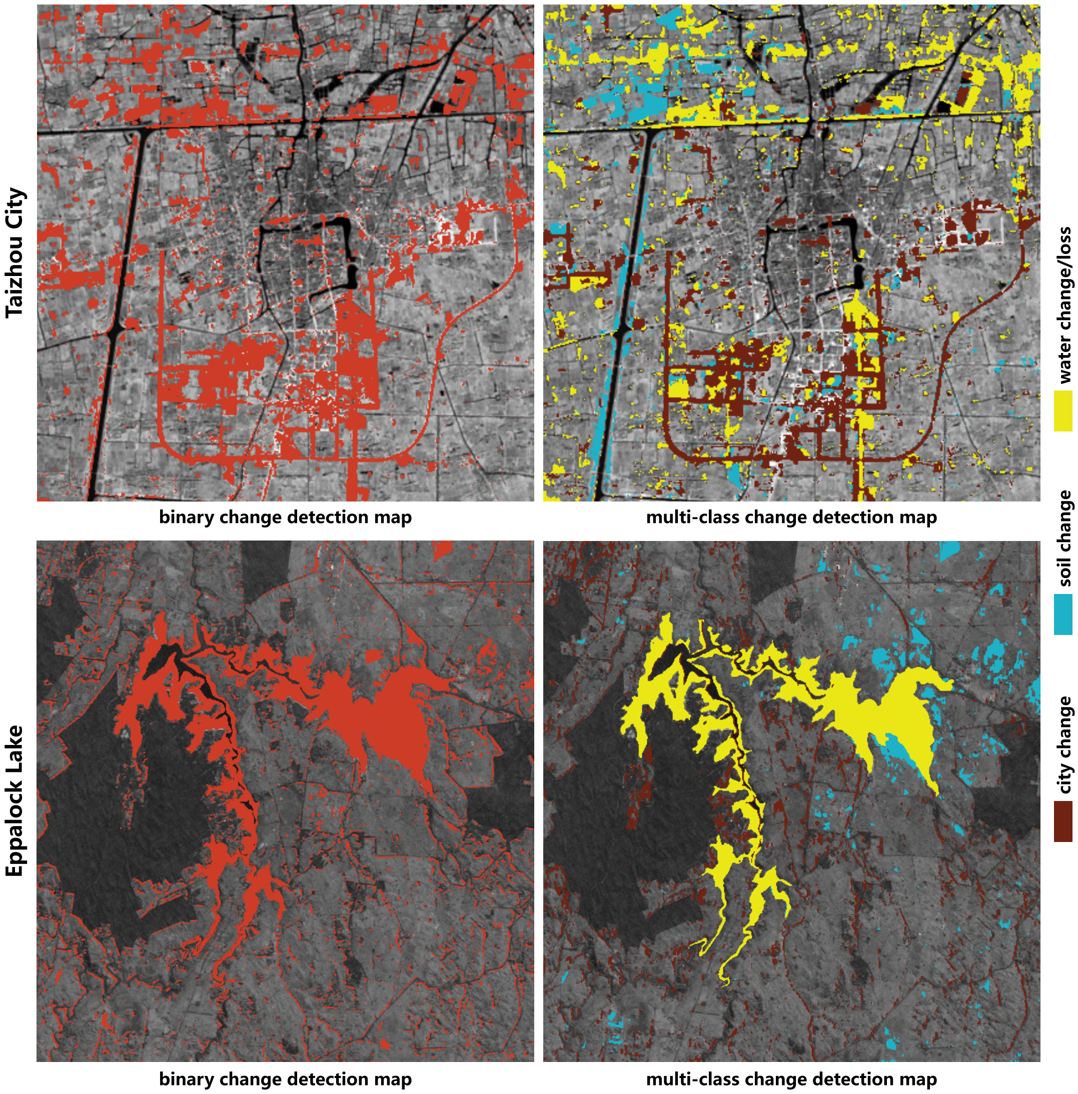

Taizhou City
binary change detection map
multi-class change detection map
Eppalock Lake
binary change detection map
multi-class change detection map
water change/loss
soil change
city change

Fig. 8. Change-detection maps generated by the proposed ReCNN-LSTM model.

control the information flow. Therefore, the total number of parameters in GRU is reduced by about 25% compared to that in LSTM. Fig. 7 shows the number of total trainable parameters in different RNN architectures.

Tables III and IV list binary and multiclass change detection results obtained in our experiments, respectively. For both data sets, ReCNN-LSTM outperforms ReCNN-FC and ReCNN-GRU on all indexes (i.e., OA and Kappa coefficient).

For example, in the binary change detection, ReCNN-LSTM increases the accuracy by 0.38% of OA and 0.0122 of Kappa on the Taizhou data set, in comparison with ReCNN-FC; by 0.06% of OA and 0.0021 of Kappa on the same data set, compared to ReCNN-GRU. However, we can see that on these data sets, all three variations of the proposed ReCNN perform closely to each other. On the other hand, the proposed networks with gating RNN architectures as the recurrent subnetwork (ReCNN-LSTM and ReCNN-GRU) slightly outperforms the more traditional ReCNN-FC on both of data sets and change detection tasks.

D. Analysis of Spatial Component: RNN versus ReCNN-LSTM

In the case of spectral-spatial-temporal change detection, the proposed recurrent convolutional network is able to significantly improve the spectral-temporal-based RNN model. As shown in Table III, compared to RNN, ReCNN-LSTM increases the accuracy of binary change detection considerably by 2.23% of OA and 0.0708 of Kappa coefficient, respectively, on the Taizhou data set. For the Eppalock lake scene, the accuracy increments on OA and Kappa coefficient are 3.46% and 0.071, respectively. Table IV compares the performance of RNN and ReCNN-LSTM in terms of multiclass change detection task. The latter can improve the former by 2.56% of OA and 0.0905 of Kappa coefficient, respectively, on the Taizhou scene; by 2.36% of OA and 0.036 of Kappa, respectively, on the Eppalock lake data. These results reveal the fact that the usage of the spatial cue in our model can construct a more powerful spectral-spatial-temporal change detector.

Furthermore, as shown in Fig. 9, it is obvious that the spectral-temporal change detection method (RNN) always results in noisy scatter points in the change detection map. However, our spectral-spatial-temporal model ReCNN-LSTM addresses this problem by eliminating noisy scattered points of wrong detection.

E. Comparison With Other Approaches

The OAs and Kappa coefficients of all competitors and the proposed networks on binary change detection task can be found in Table III. The classical change detection algorithms, CVA, PCA, MAD, and IRMAD, all achieve a good performance, especially IRMAD, which has the best performance among them. Compared to IRMAD, improvements in OA and Kappa coefficient achieved by ReCNN-LSTM are 3.59% and 0.1279, respectively, on the Taizhou data set, and increments of OA and Kappa obtained by ReCNN-LSTM on the Eppalock lake scene are 7.4% and 0.1554, respectively. However, the cost of such accuracy improvements is that we have to manually label some training data for supervised learning.

Table IV presents accuracy indexes on multiclass change detection task. Analysis of the detection accuracies indicates that SVM with RBF kernel outperforms DT, mainly because the kernel SVM generally handles nonlinear inputs more efficiently than DT. It can be seen that the proposed recurrent convolutional network ReCNN-LSTM outperforms SVM and RNN in terms of OA and Kappa coefficient on both the Taizhou and Eppalock lake data. Compared to SVM and RNN, ReCNN-LSTM increases OA by 4.14% and 2.56%, respectively, on the Taizhou data set; by 2.84% and 2.36%, respectively, on the Eppalock lake data.

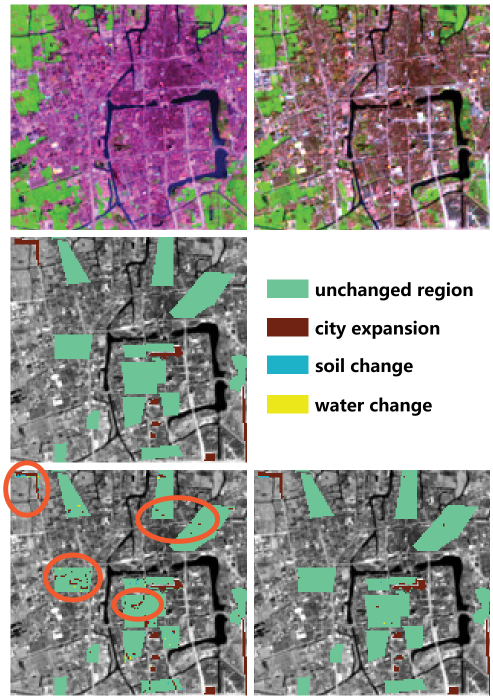


Fig. 9. Comparison between spectral–temporal model (RNN) and spectral–spatial–temporal method (ReCNN-LSTM) on a region of the Taizhou city. (Left to Right and Top to Bottom): T_1 image, T_2 image, GT, change detection map obtained from RNN, and change detection map produced by ReCNN-LSTM. It can be clearly seen that there are a number of noisy scatter points of wrong detection (see ellipses in the bottom left image) in the change detection map of RNN. While our spectral–spatial–temporal model ReCNN-LSTM addresses this problem by eliminating those points.

Fig. 8 shows change detection results of the Taizhou city and Eppalock lake obtained by our model.

IV. Conclusion

In this paper, we have proposed a novel neural network architecture, called ReCNN, which integrates the merits of both CNN and RNN. ReCNN is capable of extracting joint spectral–spatial–temporal features from bitemporal multispectral images and predicts change types. Moreover, it is end-to-end trainable. All these properties make ReCNN an excellent approach for multitemporal remote sensing data analysis.

The experiments on real multispectral images demonstrate that ReCNN achieves competitive performance, compared with conventional change detection models as well as spectral–temporal-based RNN algorithm. This confirms the advantages of the proposed recurrent convolutional network. In addition, ReCNN is a general framework; therefore, it can be applied to other domains and problems (such as multitemporal hyper spectral/multispectral data classification) that involve sequence prediction in remote sensing sequence data. Moreover, it is worth noting that the proposed network architecture has the potential to be extended and used to multisource change detection tasks. Because compared to CNN, Siamese convolutional network, and RNN, the separate yet identical convolutional branches of our network allow the network to learn different data-driven feature representations from different types of data which are usually considered to lie on various data manifolds.

Future works will focus on new architectures based on ReCNN, for example, a semisupervised ReCNN that can also use arbitrary amounts of unlabeled data for training—typically a small amount of labeled data with a large amount of unlabeled data.

Acknowledgment

The authors would like to thank Dr. C.Wu for providing the Taizhou data set.

References

[1] F. Bovolo and L. Bruzzone, "The time variable in data fusion: A change detection perspective," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 8–26, Sep. 2015.

[2] N. Yokoya, X. X. Zhu, and A. Plaza, "Multisensor coupled spectral unmixing for time-series analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2842–2857, May 2017.

[3] J. Yang, P. J. Weisberg, and N. A. Bristow, "Landsat remote sensing approaches for monitoring long-term tree cover dynamics in semi-arid woodlands: Comparison of vegetation indices and spectral mixture analysis," *Remote Sens. Environ.*, vol. 119, pp. 62–71, Apr. 2012.

[4] G. Xian and C. Homer, "Updating the 2001 national land cover database impervious surface products to 2006 using Landsat imagery change detection methods," *Remote Sens. Environ.*, vol. 114, no. 8, pp. 1676–1686, Aug. 2010.

[5] B. Liang and Q. Weng, "Assessing urban environmental quality change of Indianapolis, United States, by the remote sensing and GIS integration," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 1, pp. 43–55, Mar. 2011.

[6] W. A. Malila, "Change vector analysis: An approach for detecting forest changes with Landsat," in *Proc. Mach. Process. Remotely Sensed Data Symp.*, 1980.

[7] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 1, pp. 218–236, Jan. 2007.

[8] F. Bovolo, S. Marchesi, and L. Bruzzone, "A framework for automatic and unsupervised detection of multiple changes in multitemporal images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2196–2212, May 2012.

[9] M. Zanetti, F. Bovolo, and L. Bruzzone, "Rayleigh-Rice mixture parameter estimation via EM algorithm for change detection in multispectral images," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5004–5016, Dec. 2015.

[10] J. S. Deng, K. Wang, Y. H. Deng, and G. Qi, "PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data," *Int. J. Remote Sens.*, vol. 29, no. 16, pp. 4823–4838, 2008.

[11] J. B. Collins and C. E. Woodcock, "An assessment of several linear change detection techniques for mapping forest mortality using multitemporal Landsat TM data," *Remote Sens. Environ.*, vol. 56, no. 1, pp. 66–77, 1996.

[12] A. A. Nielsen, K. Conradsen, and J. J. Simpson, "Multivariate alteration detection (MAD) and MAF postprocessing in multispectral, bitemporal image data: New approaches to change detection studies," *Remote Sens. Environ.*, vol. 64, no. 1, pp. 1–19, 1998.

[13] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2858–2874, May 2014.

[14] A. Ertürk, M.-D. Iordache, and A. Plaza, "Sparse unmixing-based change detection for multitemporal hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 708–719, Feb. 2016.

[15] M. Gong, T. Zhan, P. Zhang, and Q. Miao, "Superpixel-based difference representation learning for change detection in multispectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2658–2673, May 2017.

[16] X. Li and A. G. O. Yeh, "Principal component analysis of stacked multi-temporal images for the monitoring of rapid urban expansion in the Pearl River Delta," *Int. J. Remote Sens.*, vol. 19, no. 8, pp. 1501–1518, 1998.

[17] L. Bruzzone and S. B. Serpico, "An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 4, pp. 858–867, Jul. 1997.

[18] L. Bruzzone, D. F. Prieto, and S. B. Serpico, "A neural-statistical approach to multitemporal and multisource remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1350–1359, May 1999.

[19] L. Bruzzone and R. Cossu, "A multiple-cascade-classifier system for a robust and partially unsupervised updating of land-cover maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 9, pp. 1984–1996, Sep. 2002.

[20] L. Bruzzone, R. Cossu, and G. Vernazza, "Detection of land-cover transitions by combining multidate classifiers," *Pattern Recogniton Lett.*, vol. 25, no. 13, pp. 1491–1500, 2004.

[21] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[22] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, p. 506, 2016.

[23] M. Rußwurm and M. Körner, "Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshop*, Jul. 2017, pp. 1496–1504.

[24] L. Mou and X. X. Zhu. (2018). "IM2HEIGHT: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network." [Online]. Available: https://arxiv.org/abs/1802.10249

[25] J. Hu, L. Mou, A. Schmitt, and X. X. Zhu, "FusioNet: A two-stream convolutional neural network for urban scene classification using PolSAR and hyperspectral data," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, Mar. 2017, pp. 1–4.

[26] L. Mou, M. Schmitt, Y. Wang, and X. X. Zhu, "A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, Mar. 2017, pp. 1–4.

[27] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2016.

[28] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.

[29] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[30] L. Mou and X. X. Zhu, "Spatiotemporal scene interpretation of space videos via deep neural network and tracklet analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1823–1826.

[31] L. H. Hughes, M. Schmitt, L. Mou, Y. Wang, and X. X. Zhu. (2018). "Identifying corresponding patches in SAR and optical images with a pseudo-Siamese CNN." [Online]. Available: https://arxiv.org/abs/1801.08467

[32] L. Mou *et al.*, "Multitemporal very high resolution from space: Outcome of the 2016 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3435–3447, Aug. 2017.

[33] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[34] L. Fang, N. He, S. Li, P. Ghamisi, and J. A. Benediktsson, "Extinction profiles fusion for hyperspectral images classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1803–1815, Mar. 2018.

[35] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018, doi: 10.1109/LGRS.2018.2830403.

[36] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "Remote sensing scene classification using multilayer stacked covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2018.2845668.

[37] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[38] Y. Hua, L. Mou, and X. X. Zhu. (2018). "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification." [Online]. Available: https://arxiv.org/abs/1807.11245

[39] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 4, p. 129, 2018.

[40] H. Wu and S. Prasad, "Convolutional recurrent neural networks for hyperspectral data classification," *Remote Sens.*, vol. 9, no. 3, p. 298, 2017.

[41] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 18, no. 7, pp. 1527–1554, 2016.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, 2015.

[44] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[45] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. (2016). "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs." [Online]. Available: https://arxiv.org/abs/1606.00915

[46] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets: Time-Frequency Methods Phase Space*. 1989.

[47] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1743–1751.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[49] A. Graves. (2013). "Generating sequences with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1308.0850

[50] T. Dozat. *Incorporating Nesterov Momentum Into Adam*. [Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf

[51] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. IEEE Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1–9.

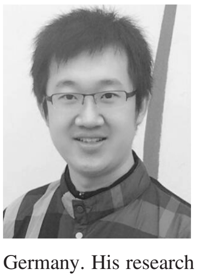
[52] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[53] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, 2015.

[54] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.

[55] A. A. Nielsen, "The regularized iteratively reweighted mad method for change detection in multi- and hyperspectral data," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 463–478, Feb. 2007.

[56] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jan. 2015, Art. no. 258619.



Lichao Mou (S'16) received the bachelor's degree in automation from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2012, and the master's degree in signal and information processing from the University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the German Aerospace Center, Germany, and the Technical University of Munich, Munich, Germany.

In 2015, he was with the Computer Vision Group, University of Freiburg, Freiburg in Breisgau, Germany. His research interests include remote sensing, computer vision, and machine learning, especially remote sensing video analysis and deep networks with their applications in remote sensing.

Dr. Mou was a recipient of the 2016 IEEE GRSS Data Fusion Contest (first place) and the Finalist for the Best Student Paper Award at the 2017 Joint Urban Remote Sensing Event.

Lorenzo Bruzzone (S'95–M'98–SM'03–F'10) received the M.S. degree *(summa cum laude)* in electronic engineering and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1993 and 1998, respectively.

He is currently a Full Professor of telecommunications with the University of Trento, Trento, Italy, where he teaches remote sensing, radar, and digital communications. He is the Founder and the Director of the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento. He is the Principal Investigator of many research projects, including *Radar for Icy Moon Exploration* instrument in the framework of the *JUpiter ICy moons Explorer* mission of the European Space Agency. He has authored or co-authored 218 scientific publications in referred international journals (157 in the IEEE journals), more than 290 papers in conference proceedings, and 21 book chapters. He has edited or co-edited 18 books or conference proceedings and 1 scientific book. His papers have been cited more than 25 000 times, h-index 74. His research interests include remote sensing, radar and SAR, signal processing, machine learning, and pattern recognition. He promotes and supervises research on these topics within the frameworks of many national and international projects.

Dr. Bruzzone was a Distinguished Speaker of the IEEE Geoscience and Remote Sensing Society from 2012 to 2016. He was invited as a Keynote Speaker in more than 30 international conferences and workshops. He is a member of the Permanent Steering Committee of this series of workshops. He has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing since 2003 and a member of the Administrative Committee of the IEEE Geoscience and Remote Sensing Society since 2009. He is the Co-Founder of the IEEE International Workshop on the Analysis of Multitemporal Remote-Sensing Images (MultiTemp) series. He was a recipient of the i Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (first place), Seattle, in 1998, many International and National Honors and Awards including the recent IEEE GRSS 2015 Outstanding Service Award, and the 2017 IEEE IGARSS Symposium Prize Paper Award. He has been the Founder of the IEEE GEOSCIENCE AND REMOTE SENSING MAGAZINE for which he has been the Editor-in-Chief from 2013 to 2017. He was a Guest Co-Editor of many special issues of international journals. He is currently an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Xiao Xiang Zhu (S'10–M'12–SM'14) received the M.Sc., Dr.-Ing., and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She was a Guest Scientist or a Visiting Professor with the Italian National Research Council (CNR-IREA), Naples, Italy, in 2009, Fudan University, Shanghai, China, in 2014, the University of Tokyo, Tokyo, Japan, in 2015, and the University of California, Los Angeles, CA, USA, in 2016. She is currently a Professor of signal processing with the Earth Observation, TUM and German Aerospace Center, Germany, the Head of the EO Data Science Department, German Aerospace Center's Earth Observation Center, German Aerospace Center, Germany, and the Head of the Helmholtz Young Investigator Group SiPEO, TUM. Her research interests include remote sensing and Earth observation, signal processing, machine learning, and data science, especially global urban mapping.

Dr. Zhu is a member of the Young Academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities, the German National Academy of Sciences Leopoldina, and the Bavarian Academy of Sciences and Humanities. She is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

D Mou L., Zhu X. 2019. Learning to Pay Attention on Spectral Domain: A Spectral Attention Module-Based Convolutional Network for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 1, pp. 110-122, 2020.

Learning to Pay Attention on Spectral Domain: A Spectral Attention Module-Based Convolutional Network for Hyperspectral Image Classification

Lichao Mou, *Student Member, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*— Over the past few years, hyperspectral image classification using convolutional neural networks (CNNs) has progressed significantly. In spite of their effectiveness, given that hyperspectral images are of high dimensionality, CNNs can be hindered by their modeling of all spectral bands with the same weight, as probably not all bands are equally informative and predictive. Moreover, the usage of useless spectral bands in CNNs may even introduce noises and weaken the performance of networks. For the sake of boosting the representational capacity of CNNs for spectral-spatial hyperspectral data classification, in this work, we improve networks by discriminating the significance of different spectral bands. We design a network unit, which is termed as the spectral attention module, that makes use of a gating mechanism to adaptively recalibrate spectral bands by selectively emphasizing informative bands and suppressing less useful ones. We theoretically analyze and discuss why such a spectral attention module helps in a CNN for hyperspectral image classification. We demonstrate using extensive experiments that in comparison with state-of-the-art approaches, the spectral attention module-based convolutional networks are able to offer competitive results. Furthermore, this work sheds light on how a CNN interacts with spectral bands for the purpose of classification.**

***Index Terms*— Attention module, convolutional neural network (CNN), gating mechanism, hyperspectral image classification.**

I. Introduction

HYPERSPECTRAL images encompass hundreds of continuous observation spectral bands, which are capable of precisely differentiating various materials of interest. Hence, in the remote sensing community, hyperspectral images have already been considered a vital data source for object identification and classification tasks.

Consequently, numerous kinds of classification approaches, especially supervised models have been developed for hyperspectral data classification, as found in the literature. Among them, random forest [1]–[3] and support vector machine (SVM) [4]–[8] are two examples of supervised classification approaches, which have been exploited for solving varied and numerous classification problems. Random forests are basically a kind of ensemble bagging or averaging algorithm. It creates a set of decision trees using random subsamples of training data and then aggregates their predictions via a maximum a posterior (MAP) rule or voting to decide final classes of test samples. On the other hand, an SVM seeks for a hyperplane that is able to sort two-class data by the largest margin. However, the random forest and SVM are characterized as "shallow" models [9] as compared to deep networks which are able to extract hierarchical, deep feature representations.

Deep learning, which is mainly characterized by deep networks, has been quite successful in solving a wide range of problems (e.g., natural language processing [10]–[12], computer vision [13]–[25], and remote sensing [26]). In the hyperspectral community, some studies have been published recently on the use of convolutional neural networks (CNNs) [27]–[42] as well as recurrent neural networks (RNNs) [43]–[49] for pattern recognition tasks. For instance, Kussul *et al.* [27] addressed the classification problem of crop types by making use of 1-D and 2-D CNNs and found that the 2-D CNN is superior to the 1-D CNN, but several tiny objects in the classification map of the 2-D CNN are a little oversmoothed and misclassified. In [28], Song *et al.* studied feature fusion in a residual learning-based 2-D CNN, aiming to build a more discriminative network for hyperspectral data classification tasks. Following the recent developments in 3-D CNN for video analysis [50], where the third dimensionality is usually the time axis, 3-D CNNs have also been studied in hyperspectral data classification. Chen *et al.* [29] introduced a ℓ_2 regularized 3-D CNN for learning spectral-spatial features, while [30] followed a similar idea for the purpose of classification. Paoletti *et al.* [51] introduced an improved 3-D CNN consisting of 5 layers which make use of all the spatial-spectral information on the hyperspectral image.

To avoid overfitting, Zhao and Du [32] jointly used a dimension reduction method and a 2-D CNN for spectral-spatial

Manuscript received December 27, 2018; revised May 5, 2019; accepted June 23, 2019. Date of publication September 27, 2019; date of current version December 27, 2019. This work was supported in part by the German Research Foundation (DFG) under Grant ZH 498/7-2, in part by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Program (Acronym: *So2Sat*, www.so2sat.eu) under Grant [ERC-2016-StG-714087], and in part by the Helmholtz Association under the framework of the Young Investigators Group "SiPEO" (www.sipeo.bgu.tum.de) under Grant VH-NG-1018. *(Corresponding author: Xiao Xiang Zhu.)*

The authors are with the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), 82234 Wessling, Germany, and also with the Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: lichao.mou@dlr.de; xiaoxiang.zhu@dlr.de).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2933609

feature extraction. Ghamisi *et al.* [33] first exploited a computational intelligence (particle swarm optimization) method to choose informative spectral bands and then train a 2-D CNN using the selected bands. In [34], to properly train a CNN with limited ground truth data, the authors devised a pixel-pair CNN that takes as input a pair of hyperspectral pixels. By doing so, the amount of training data is greatly augmented. Furthermore, in order to access a huge amount of unlabeled hyperspectral data, unsupervised feature learning via a CNN is of great interest. Romero *et al.* [35] presented a CNN to address the problem of unsupervised spectral-spatial feature extraction and estimated network weights via a sparse learning approach in a greedy layer-wise fashion. Mou *et al.* [37] proposed a residual learning-based fully conv-deconv network, aiming at unsupervised spectral-spatial feature learning in an end-to-end manner. Better classification network architecture from computer vision (e.g., ResNet [17], DenseNet [18], and CapsuleNet [52]) also provides new insights into hyperspectral image classification [37]–[39], [53]. Moreover, the integration of networks and other traditional machine learning models, e.g., conditional random field (CRF) and active learning, has also received attention recently [54], [55].

The unique asset of hyperspectral images is their rich spectral content in comparison with high-resolution aerial images and natural images in the computer vision field. Although there already exist a number of works that have focused on using CNNs for hyperspectral data classification, we notice that in the community, the following questions have not been well addressed until now.

1) Do all spectral bands contribute equally to *a CNN* for classification tasks?
2) If no, how to *task-drivenly* find informative bands that can help hyperspectral data classification *in an end-to-end network*?
3) Is it possible to improve classification results of a CNN by emphasizing informative bands and suppressing less useful ones in the network?

These questions give us an incentive to devise a novel network called spectral attention module-based convolutional network for hyperspectral image classification. Inspired by recent advances in the attention mechanism of networks [56]–[58], which enables feature interactions to contribute differently to predictions, we design a channel attention mechanism for analyzing the significance of different spectral bands and recalibrating them. More importantly, the significance analysis is automatically learned from tasks and hyperspectral data in an end-to-end network without any human domain knowledge. Experiments show that the use of the proposed spectral attention module in a CNN for hyperspectral data classification serves two benefits: it not only offers better performance but also provides an insight into which spectral bands contribute more to predictions. This work's contributions are threefold.

1) We propose a learnable spectral attention module that explicitly allows the spectral manipulation of hyperspectral data within a CNN. This attention module exploits the global spectral-spatial context for producing a series of spectral gates which reflects the significance of spectral bands. The recalibrated spectral information using these spectral gates can effectively improve the classification results.
2) We analyze and discuss why the proposed spectral attention module is able to offer better classification results from a theoretical perspective by diving into the backward propagation of the network. As far as we know, learning and analyzing such a spectral attention-based network for hyperspectral image classification have not been done yet.
3) We conduct experiments on four benchmark data sets. The empirical results demonstrate that our spectral attention module-based convolutional network is capable of offering competitive classification results, particularly in the situation of high dimensionality and inadequate training data.

The remainder of this article is organized as follows. After detailing hyperspectral image classification using CNNs in Section I, Section II introduces the proposed spectral attention module-based convolutional network. Section III verifies the proposed approach and presents the corresponding analysis and discussion. Finally, Section IV concludes the article.

II. Methodology

A. Problem Formulation

The spectral attention module in our model transforms a patch $\boldsymbol{x}$ of a hyperspectral image into a new representation $\boldsymbol{z}$ via the following mapping:

$$\boldsymbol{F} : \boldsymbol{x} \to \boldsymbol{z} \tag{1}$$

where $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^{H \times W \times C}$.

Our aim is to strengthen the representational capacity of a spectral-spatial classification network through explicitly modeling the significance of spectral bands. Therefore, we instantiate $\boldsymbol{F}$ as

$$\boldsymbol{z} = \boldsymbol{x} \odot \boldsymbol{g} \tag{2}$$

where $\odot$ is a channel-wise multiplication operation and $\boldsymbol{g} \in \mathbb{R}^{C}$ represents a set of *spectral gates* applied to individual spectral bands of the patch $\boldsymbol{x}$.

The motivation behind (2) is that we wish to make use of a gating mechanism to recalibrate the strength of different spectral bands of the input, i.e., selectively emphasize useful bands and suppress less informative ones, for image classification problems.

Fig. 1 illustrates the architecture of the spectral attention module-equipped convolutional network.

B. Modeling of Spectral Attention Module

The gating mechanism has been widely used in modeling and processing temporal sequences. For example, long short-term memory (LSTM)-based networks [59], [60] harness three gates to cope with vanishing gradients. Similarly, a gated recurrent unit (GRU) [61], [62] is designed to implement the modulation of information flow through the gating mechanism.

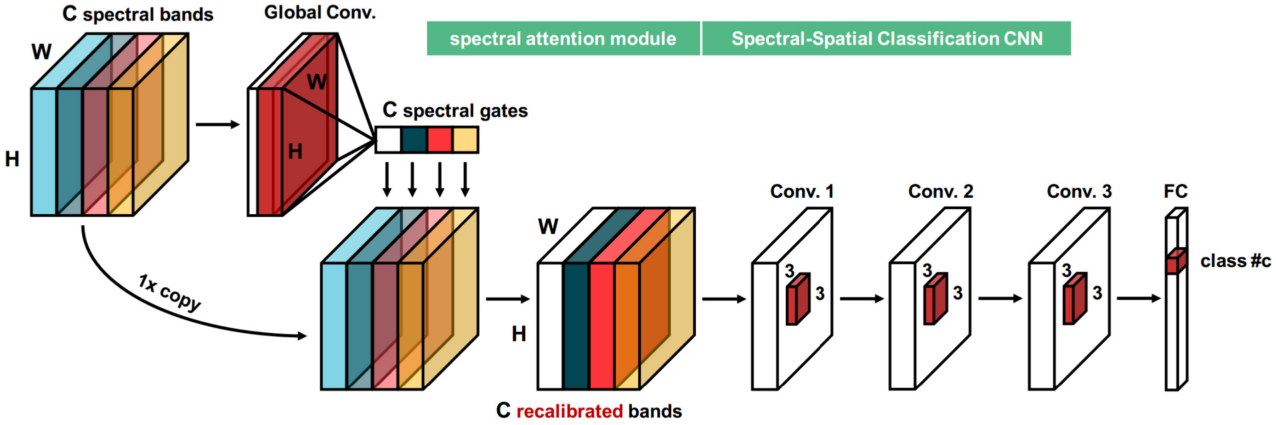


Fig. 1. The overall architecture of the proposed gating mechanism, spectral attention module, for hyperspectral classification problems. We would like to exploit this module to learn and recalibrate strengths of different spectral bands, i.e., selectively emphasize useful bands and suppress less informative ones, for image classification problems. To this end, we first learn a set of spectral gates by using global convolution and then apply them to individual spectral bands. Moreover, in Section II-C, we theoretically analyze and discuss why the proposed spectral attention module can help a spectral-spatial classification network (e.g., a 2-D CNN) for hyperspectral image classification tasks.

In addition, several recent works in computer vision have shown the benefit of introducing the gating mechanism to vision problems. To name a few, Wang *et al.* [56] proposed a gating mechanism that is capable of dynamically balancing contributions of the current event and its surrounding contexts in their model for dense video captioning tasks. Hu *et al.* [58] built a gated block for image classification tasks and demonstrated its good performance on large-scale image recognition. Liu *et al.* [57] addressed person re-identification tasks through utilizing a network module based on a soft gating mechanism, which enables the network to concentrate on significant local regions of an input image pair adaptively. In remote sensing, a very recently published, parallel work related to this article can be found in [63], where the authors introduced a visual attention technique that first calculates a mask and then applies it to features produced by a ResNet for hyperspectral data classification tasks.

Here, we would like to design our own gating mechanism, spectral attention module, for analyzing the significance of different spectral bands and recalibrating them. Besides, we hope this module is task-driven and can be adaptively learned in an end-to-end spectral-spatial classification network. To this end, we need a way to aggregate the spectral-spatial information of $\boldsymbol{x}$ across the spatial domain to produce a collection of spectral gates $\boldsymbol{g}$.

The convolution operation is an ideal candidate, as 1) it is able to spatially shrink the input patch and 2) its differential property allows end-to-end learning. In general, a convolutional filter operates with a local receptive field (e.g., 3 × 3 in VGG-16 network), which leads to the fact that the output is not capable of utilizing contextual information outside of this region. This is a severe issue for our case because the spectral gates $\boldsymbol{g}$ in our model are expected to be derived from the whole spectral-spatial information. To tackle this problem, we distill global spatial information into the spectral gates by using global convolution. Formally, let $\boldsymbol{f} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_C]$ denote a set of convolutional filters and their sizes are both $H \times W$, where $\boldsymbol{f}_c$ refers to the c-th filter. Thus, the c-th spectral gate g_c can be calculated as follows:

$$g_c = \boldsymbol{x} * \boldsymbol{f}_c = \sum_{i=1}^{C} \boldsymbol{x}_i * \boldsymbol{f}_c^i \tag{3}$$

where $*$ represents convolution and $\boldsymbol{f}_c^i$ and $\boldsymbol{x}_i$ are separately the i-th channels of the c-th filter and $\boldsymbol{x}$. Taking into account that the field of view of global convolution is equal to the spatial size of $\boldsymbol{x}$, g_c is actually calculated by the inner product of $\boldsymbol{x}_i$ and $\boldsymbol{f}_c^i$ (both $\boldsymbol{x}_i$ and $\boldsymbol{f}_c^i$ are vectorized into columns), i.e., (3) can be rewritten as follows:

$$g_c = \sum_{i=1}^{C} \langle \boldsymbol{x}_i, \boldsymbol{f}_c^i \rangle = \sum_{i=1}^{C} \boldsymbol{x}_i^T \boldsymbol{f}_c^i. \tag{4}$$

From (4), the spectral gates $\boldsymbol{g}$ can be considered as a series of global descriptors, which are capable of representing spectral-spatial features of $\boldsymbol{x}$.

Thus, according to (2), we can associate the c-th spectral gate g_c with the c-th spectral band of $\boldsymbol{x}$ to obtain the recalibrated z_c via

$$z_c = \boldsymbol{x}_c \sum_{i=1}^{C} \boldsymbol{x}_i^T \boldsymbol{f}_c^i. \tag{5}$$

So far, we can obtain an initial spectral attention module [as shown in (5)], but there still exist three issues which we should address:

1) Given the complex spectral-spatial properties of hyperspectral images, we wish that the spectral gates in this module are capable of learning a nonlinear mapping, instead of a linear one, from the input.
2) The attention module should model a nonmutually exclusive relationship between spectral bands, as we would like to ensure that multiple bands can be emphasized at the same time (unlike one-hot activation in softmax).

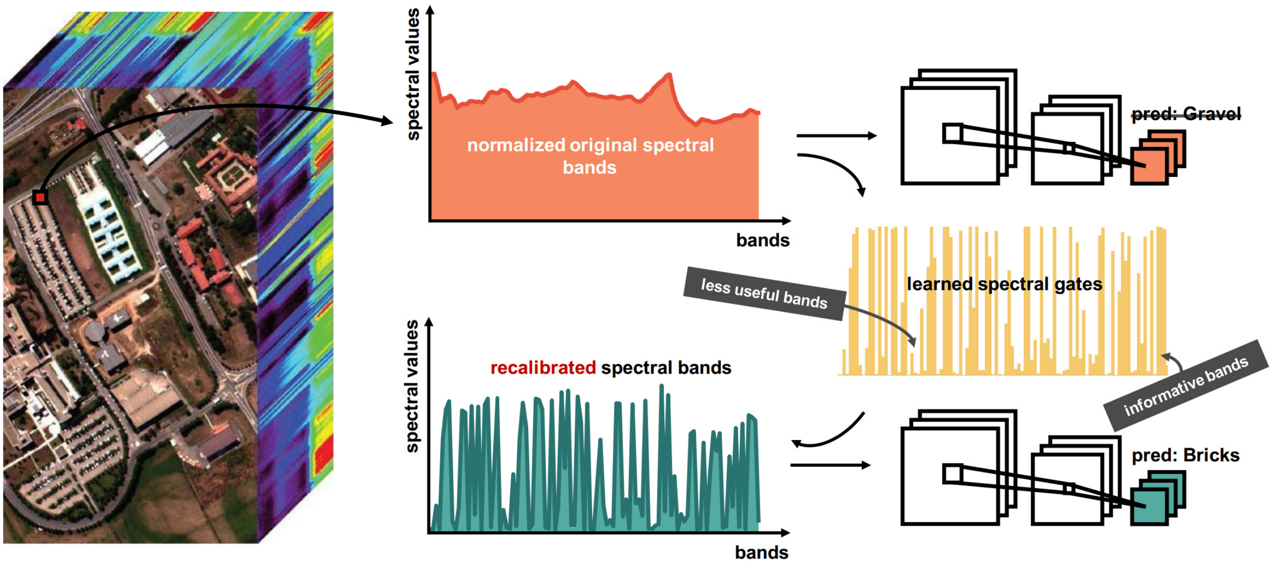


Fig. 2. Example showing how the proposed spectral attention module in a CNN correct a wrong prediction (gravel) to a right one (bricks) via learned spectral gates.

3) The gates should be bounded (e.g., between 0 and 1), easily differentiable, and monotonic (good for convex optimization).

To meet these three requirements, we modify spectral gates in the initial spectral attention module as follows:

$$\begin{aligned} g_c &= \frac{1}{1+\exp(-\boldsymbol{x} * \boldsymbol{f}_c)} \\ &= \frac{1}{1+\exp\left(-\sum_{i=1}^{C} \boldsymbol{x}_i^T \boldsymbol{f}_c^i\right)}. \end{aligned} \tag{6}$$

Hence, the final version of the spectral attention module can be written as

$$z_c = \boldsymbol{x}_c \frac{1}{1+\exp\left(-\sum_{i=1}^{C} \boldsymbol{x}_i^T \boldsymbol{f}_c^i\right)}. \tag{7}$$

Fig. 2 is an example showing how the proposed spectral attention module works in a CNN.

C. Why Does the Spectral Attention Module Work?

In our experiments, we observed that a 2-D CNN with our spectral attention module can offer better classification results. However, how exactly does this attention module help a spectral-spatial classification network for hyperspectral data classification? We dive into the backward propagation of the network to seek the answer to this question.

For notional simplicity, we subsequently drop the subscript c and rewrite the final expression of the spectral attention module as follows:

$$z = \boldsymbol{x} \odot \frac{1}{1+\exp(-\boldsymbol{x} * \boldsymbol{f})}. \tag{8}$$

Thus, the gradient of the spectral attention module can be written as

$$\begin{aligned} \nabla z = \nabla \boldsymbol{x} \odot \frac{1}{1+\exp(-\boldsymbol{x} * \boldsymbol{f})} \\ + \boldsymbol{x} \odot \nabla \left(\frac{1}{1+\exp(-\boldsymbol{x} * \boldsymbol{f})} \right). \end{aligned} \tag{9}$$

It can be seen that the term $\nabla \boldsymbol{x}$ is weighted by the spectral gates $(1/1+\exp(-\boldsymbol{x} * \boldsymbol{f}))$. This has the following interesting properties.

1) On the one hand, the existence of $\nabla \boldsymbol{x}$ ensures that the gradient information on spectral-spatial features can be backpropagated directly, which helps to prevent the vanishing gradient problem.
2) On the other hand, for spectral bands where the spectral gates are close to 0 (less useful bands), the gradient propagation vanished; on the contrary, for values that are close to 1, gradients (of informative bands) directly propagated from $\boldsymbol{z}$ to $\boldsymbol{x}$.

For the first point, a similar effect can be found in residual learning. He *et al.* [17] introduced the residual learning into CNNs for large-scale image classification tasks and exhibited significantly improved network training characteristics, e.g., allowing network depths that were previously unattainable. Formally, denote by $\boldsymbol{y}$ a random variable representing the output of a residual block. It can then be expressed as

$$\boldsymbol{y} = \boldsymbol{x} + \mathcal{F}(\boldsymbol{x}; \boldsymbol{w}) \tag{10}$$

where $\mathcal{F}$ is a residual function and usually implemented by a couple of stacked convolutional layers. Moreover, $\boldsymbol{w}$ represents learnable weights of this residual block. The gradient of a residual block can be calculated as

$$\nabla \boldsymbol{y} = \nabla \boldsymbol{x} + \nabla(\mathcal{F}(\boldsymbol{x}; \boldsymbol{w})). \tag{11}$$

TABLE I

CONFIGURATION OF A SPECTRAL ATTENTION MODULE-BASED CONVOLUTIONAL NETWORK FOR THE PAVIA UNIVERSITY DATA SET

Layer	Input Shape	Output Shape	#Filters	Connected to	Configuration
spec. attn. module	(16, 16, 103)	(16, 16, 103)	103	input	16×16 kernel
conv1-1	(16, 16, 103)	(16, 16, 32)	32	spec. attn. module	3 × 3 kernel, stride 1, padding 1, bn, relu
conv1-2	(16, 16, 32)	(16, 16, 32)	32	conv1-1	3 × 3 kernel, stride 1, padding 1, bn, relu
maxpool1	(16, 16, 32)	(8, 8, 32)	-	conv1-2	pool size 2 × 2, stride 2
conv2-1	(8, 8, 32)	(8, 8, 64)	64	maxpool1	3 × 3 kernel, stride 1, padding 1, bn, relu
conv2-2	(8, 8, 64)	(8, 8, 64)	64	conv2-1	3 × 3 kernel, stride 1, padding 1, bn, relu
maxpool2	(8, 8, 64)	(4, 4, 64)	-	conv2-2	pool size 2 × 2, stride 2
conv3-1	(4, 4, 64)	(4, 4, 128)	128	maxpool2	3 × 3 kernel, stride 1, padding 1, bn, relu
conv3-2	(4, 4, 128)	(4, 4, 128)	128	conv3-1	3 × 3 kernel, stride 1, padding 1, bn, relu
maxpool3	(4, 4, 128)	(2, 2, 128)	-	conv3-2	pool size 2 × 2, stride 2
gap	(2, 2, 128)	(1, 1, 128)	-	maxpool3	pool size 2 × 2
fc1	(1, 1, 128)	(1024,)	-	gap	1024 units, relu
fc2	(1024,)	(9,)	-	fc1	9 units, softmax

From (11), we can see that $\nabla \boldsymbol{y}$ is a sum of the gradient of the input $\nabla \boldsymbol{x}$ and the gradient $\nabla(\mathcal{F}(\boldsymbol{x}; \boldsymbol{w}))$, and as mentioned above, the term $\nabla \boldsymbol{x}$ is a key to avoiding the vanishing gradient problem. This is the same for the first property of our spectral attention module.

Instead of $\nabla \boldsymbol{x}$ in (9), $\nabla \boldsymbol{x}$ in (11) is not weighted – in other words, gradients of all spectral bands are indiscriminately backpropagated; in contrast, the spectral attention module has a selection mechanism regarding the significance of different spectral bands from the perspective of gradient.

D. Network Training

We insert the spectral attention module into a 2-D CNN (between the input and the first convolutional layer) and then train the whole network. Note that the spectral attention module and other layers are trained simultaneously. We use the TensorFlow framework to implement and train networks. All network weights are initialized by a Glorot uniform initializer [64]. The Nesterov Adam [65] algorithm is chosen to optimize networks, as for our experiments, compared to stochastic gradient descent (SGD) with momentum [66] or Adam [67], it is able to provide much faster convergence. Almost all parameters of this optimizer are set as recommended in [65]. We utilize a relatively small learning rate of $2e{-}04$. Finally, we train networks on an NVIDIA Tesla P100 16 GB GPU. Table I exhibits an example of a CNN with the proposed attention module.

III. EXPERIMENTS AND ANALYSIS

A. Data Description

1) Indian Pines Hyperspectral Data Set: The first data were collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over Northwest Indiana, USA, 1992. It includes 145 × 145 pixels with a 20 m/pixel spatial resolution and 200 spectral bands covering from 400 to 2500 nm after removing 20 water absorption channels (220, 150-163, and 104-108). The ground truth includes 16 classes of interest, which are mostly various crops in different growth phases and are detailed in Table II. Since these 16 classes have similar spectral signatures, the precise classification of this scene is hard. The true-color composite image and the available ground truth data can be found in Fig. 3 (black color in the ground truth indicates unknown samples).

TABLE II

AMOUNTS OF TRAINING AND TEST DATA ON THE INDIAN PINES SCENE

Class No.	Class Name	Training	Test
1	Corn-notill	50	1384
2	Corn-min	50	784
3	Corn	50	184
4	Grass-pasture	50	447
5	Grass-trees	50	697
6	Hay-windrowed	50	439
7	Soybean-notill	50	918
8	Soybean-mintill	50	2418
9	Soybean-clean	50	564
10	Wheat	50	162
11	Woods	50	1244
12	Buildings-grass-trees	50	330
13	Stone-steel-towers	50	45
14	Alfalfa	15	39
15	Grass-pasture-mowed	15	11
16	Oats	15	5
	TOTAL	695	9671

2) Pavia University Hyperspectral Data Set: The second data set was acquired over the city of Pavia, Italy, 2002 by an airborne instrument – Reflective Optics Spectrographic Imaging System (ROSIS). The aircraft was operated by the German Aerospace Center (DLR) within the context of European Union funded HySens project. The data set is made up of 640 × 340 pixels with a 1.3 m/pixel spatial resolution and 103 bands covering from 430 to 860 nm after removing 12 noisy channels. Besides unknown pixels, 9 classes are manually annotated in the reference data. Fig. 4 displays a composite image of this data set and its reference map. Table III offers information on all 9 categories.

3) Salinas Hyperspectral Data Set: The third data set was also gathered by the AVIRIS sensor over the region of Salinas Valley, CA, USA and with a 3.7-m/pixel spatial resolution.

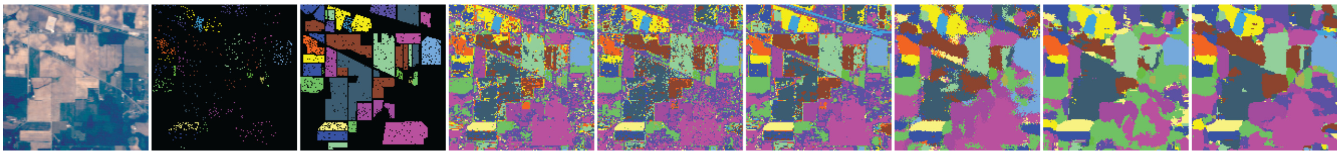


Fig. 3. Classification maps of different approaches for the Indian Pines data set. (Left to right) True-color composite image, training set, test set, RF-200, SVM-RBF, CCF-200, SICNN, 2-D CNN, and SpecAttenNet. Best zoomed-in view.

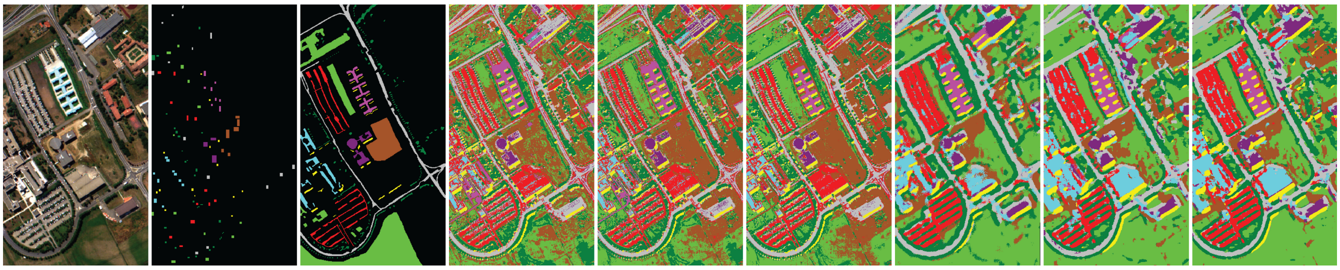


Fig. 4. Classification maps of different approaches for the Pavia University data set. (Left to right) Composite image, training samples, ground truth, RF-200, SVM-RBF, CCF-200, SICNN, 2-D CNN, and SpecAttenNet. Best zoomed-in view.

TABLE III
AMOUNTS OF TRAINING AND TEST DATA ON THE PAVIA UNIVERSITY DATA SET

Class No.	Class Name	Training	Test
1	Asphalt	548	6631
2	Meadows	540	18649
3	Gravel	392	2099
4	Trees	524	3064
5	Metal sheets	265	1345
6	Bare Soil	532	5029
7	Bitumen	375	1330
8	Bricks	514	3682
9	Shadows	231	947
	TOTAL	3921	42776

The Salinas scene is composed of 224 spectral bands and 512 × 217 pixels. Like the Indian Pines data set, 20 water absorption bands (224, 154-167, and 108-112) of the Salinas scene have been discarded. The data set presents 16 classes related to vegetables, vineyard fields, and bare soils. Table IV shows the amounts of training and test data on this data set.

4) Houston Hyperspectral Data Set: The fourth data set was acquired over the University of Houston campus and its neighboring urban area. It was collected with an ITRES-CASI 1500 sensor on June 23, 2012 between 17:37:10 and 17:39:50 UTC. The average altitude of the sensor was about 1676 m, which results in 2.5-m spatial resolution data consisting of 349 by 1905 pixels. The hyperspectral imagery consists of 144 spectral bands ranging from 380 to 1050 nm and was processed (radiometric correction, attitude processing, GPS processing, geo-correction, and so on) to yield the final geo-corrected image cube representing the sensor spectral radiance. Table V provides information about all 15 classes of this data set with their corresponding training and test samples. This data set was kindly made available by the Image Analysis and Data Fusion Technical Committee of IEEE GRSS in 2012.

TABLE IV
AMOUNTS OF TRAINING AND TEST DATA ON THE SALINAS DATA

Class No.	Class Name	Training	Test
1	Brocoli_green_weeds_1	50	1959
2	Brocoli_green_weeds_2	50	3676
3	Fallow	50	1926
4	Fallow_rough_plow	50	1344
5	Fallow_smooth	50	2628
6	Stubble	50	3909
7	Celery	50	3529
8	Grapes_untrained	50	11221
9	Soil_vinyard_develop	50	6153
10	Corn_senesced_green_weeds	50	3228
11	Lettuce_romaine_4wk	50	1018
12	Lettuce_romaine_5wk	50	1877
13	Lettuce_romaine_6wk	50	866
14	Lettuce_romaine_7wk	50	1020
15	Vinyard_untrained	50	7218
16	Vinyard_vertical_trellis	50	1757
	TOTAL	800	53329

B. Experiment Setup

To quantitatively compare different models for hyperspectral data classification tasks from various aspects, the following measurements are considered.

1) *Overall Accuracy (OA):* This criterion is calculated as the fraction of test samples that are differentiated correctly.
2) *Per-Class Accuracy:* To assess the performance with respect to each category in a data set, we also compute per-class accuracy. This measurement is particularly useful when class labels are not uniformly distributed.
3) *Average Accuracy (AA):* This criterion is computed as the average of all per-class accuracies.
4) *Kappa Coefficient:* This statistic criterion is a robustness measurement with the degree of agreement.

Furthermore, we make use of a statistical test to validate the significance of classification accuracies produced by

TABLE V
AMOUNTS OF TRAINING AND TEST DATA ON THE HOUSTON DATA SET

Class No.	Class Name	Training	Test
1	Grass Healthy	198	1053
2	Grass Stressed	190	1064
3	Grass Synthetic	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot 1	192	1041
13	Parking Lot 2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
	TOTAL	2832	12197

various methods. Given that samples used for two classification models are not independent, McNemar's test can be harnessed to estimate the significance of the difference in two classification maps, and the McNemar's test can be performed by

$$z_{12} = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \tag{12}$$

where f_{ij} is the amount of data correctly recognized by method i and incorrectly recognized by j. McNemar's test is a statistical test for paired nominal data, and we can use McNemar's test to compare the predicted accuracies of two models. In McNemar's test, the null hypothesis, which means none of the two models performs better than the other, is rejected at $p = 0.05$ ($|z| > 1.96$), which indicates the significance level.

Below are methods included in our comparison.

1) *RF-200:* A random forest composed of 200 decision trees.
2) *SVM-RBF:* An SVM[1] having the widely used radial basis function (RBF) kernel. We make use of five-fold cross validation to search optimal hyper-parameters γ (spread of the RBF kernel) and C (controlling the magnitude of penalization during the model optimization) in the range of $\gamma = 2^{-3}, 2^{-2}, \cdots, 2^{4}$ and $C = 10^{-2}, 10^{-1}, \cdots, 10^{4}$.
3) *CCF-200:* A canonical correlation forest (CCF)[2] [68], [69] with 200 trees.
4) *SICNN:* A CNN model, which makes an attempt at solving the curse of dimensionality by first utilizing a computational intelligence (particle swarm optimization) algorithm to choose informative spectral bands and then training a 2-D CNN using the selected bands. The used network is made up of three convolutional layers. The first two convolutional layers are followed by max-pooling layers and their fields of view are 4 × 4 and 5 × 5, respectively. The last convolutional layer is equipped with 4 × 4 filters. Moreover, 32, 64, and 128 convolutional filters are used separately for those three convolutional layers. For more details, refer to [33].
5) *2-D CNN:* To demonstrate the superiority of the proposed method, we perform an ablation study, i.e., designing a 2-D CNN which has no spectral attention module, but other parts are the same as the proposed network (cf. Table I). The exact architecture of the 2-D CNN is a VGG-like network, in which we utilize three convolutional blocks and 3 × 3 filters for all the blocks. Spatial shrinkage is operated by three max-pooling layers following the convolutional blocks. Each convolutional block in this 2-D CNN has two convolutional layers, and 32, 64, and 128 filters are used for convolutional layers of those three blocks, respectively. Overall, we keep the architecture of 2-D CNN and that of the following network consistent.
6) *SpecAttenNet:* The proposed spectral attention module-based convolutional network (cf. Table I).

Note that, in order to make our model completely comparable with other investigated approaches, we use standard training and test sets for the Indian Pines, Pavia University, and Houston data sets. For the Salinas scene, training samples are generated by a simple random sampling. In both hyperspectral data sets, 10% samples of the training set are randomly selected as validation samples. In other words, in the network training phase, we use 90% samples of the training set to iteratively update and optimize network weights and the remaining ones as validation to tune hyperparameters of networks. Prior to training, we normalize each channel of the hyperspectral data to the range between 0 and 1. In addition, network architecture for these data sets is the same.

C. Ablation Study

To validate the effectiveness of the proposed module, we perform ablation experiments. As we have mentioned above, the competitor 2-D CNN is a network that has no spectral attention module, but other parts are the same as the proposed SpecAttenNet. From Tables VI–IX, we can see that SpecAttenNet outperforms 2-D CNN on all indexes on all four data sets. Specifically, SpecAttenNet increases accuracies significantly by 7.46% of OA, 4.75% of AA, and 0.0849 of Kappa coefficient on the Indian Pines data set; by 2.21% of OA, 1.28% of AA, and 0.0293 of Kappa coefficient on the Pavia University data set; by 2.76% of OA, 2.87% of AA, and 0.0303 of Kappa coefficient on the Salinas scene; and by 3.1% of OA, 4.93% of AA, and 0.0333 of Kappa coefficient on the Houston scene. This shows that recalibrated spectral bands obtained by our gating mechanism become more separable for a spectral-spatial classification network, as informative bands have been emphasized, and less useful ones have been suppressed.

D. Results and Discussion

Tables VI–IX give information about per-class accuracies, OAs, AAs, and kappa coefficients obtained by various spectral

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[2] https://github.com/twgr/ccfs

TABLE VI

ACCURACY COMPARISONS FOR THE INDIAN PINES SCENE. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	SICNN	2D CNN	SpecAttenNet
1	Alfalfa	56.65	71.39	76.37	79.84	54.77	**90.46**
2	Corn-notill	55.48	71.05	77.93	92.47	**96.94**	92.22
3	Corn-min	82.07	86.96	94.57	99.46	99.46	**100**
4	Corn	85.23	91.72	94.41	93.29	**96.87**	93.96
5	Grass-pasture	80.20	85.80	91.39	92.68	94.12	**95.55**
6	Grass-trees	94.99	93.85	97.04	96.58	96.81	**99.77**
7	Grass-pasture-mowed	77.02	75.38	90.96	86.82	**91.29**	89.65
8	Hay-windrowed	57.94	59.88	69.48	69.52	**93.05**	88.79
9	Oats	62.94	76.24	**89.01**	83.69	87.59	85.64
10	Soybean-notill	95.06	96.91	98.77	**100**	**100**	**100**
11	Soybean-mintill	88.67	79.58	93.73	**96.70**	68.57	96.22
12	Soybean-clean	53.33	74.84	74.55	96.97	88.48	**98.79**
13	Wheat	97.78	97.78	**100**	**100**	**100**	**100**
14	Woods	56.41	79.49	**97.44**	94.87	82.05	94.87
15	Buildings-grass-trees	81.82	**100**	90.91	**100**	**100**	**100**
16	Stone-steel-towers	**100**	**100**	**100**	**100**	**100**	**100**
OA	-	69.31	74.24	82.87	85.13	84.76	**92.22**
AA	-	76.60	83.80	89.78	92.68	90.62	**95.37**
Kappa	-	0.6538	0.7093	0.8059	0.8313	0.8261	**0.9110**

TABLE VII

ACCURACY COMPARISONS FOR THE PAVIA UNIVERSITY SCENE. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	SICNN	2D CNN	SpecAttenNet
1	Asphalt	81.54	82.37	86.59	84.21	83.85	**86.71**
2	Meadows	55.39	67.87	72.33	91.10	96.09	**98.47**
3	Gravel	53.07	69.18	71.75	64.36	**81.47**	77.47
4	Trees	98.76	98.37	**99.09**	95.53	96.12	96.83
5	Metal Sheets	99.11	99.41	**99.78**	97.70	98.74	98.81
6	Bare Soil	79.10	93.64	**97.26**	56.53	49.79	53.11
7	Bitumen	84.36	91.20	**91.88**	77.29	79.32	77.82
8	Bricks	91.39	92.59	94.92	**95.57**	88.89	94.43
9	Shadows	97.47	96.94	**98.73**	96.20	94.19	96.30
OA	-	71.53	79.89	83.36	85.25	86.93	**89.14**
AA	-	82.24	87.95	**90.26**	84.28	85.38	86.66
Kappa	-	0.6504	0.7491	0.7905	0.8041	0.8242	**0.8535**

TABLE VIII

ACCURACY COMPARISONS FOR THE SALINAS DATA. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	2D CNN	SpecAttenNet
1	Brocoli_green_weeds_1	99.29	98.98	**99.49**	71.57	94.84
2	Brocoli_green_weeds_2	99.21	99.67	99.95	99.86	**99.97**
3	Fallow	97.72	98.70	99.43	88.89	**99.64**
4	Fallow_rough_plow	97.62	97.77	**99.33**	98.14	98.88
5	Fallow_smooth	97.95	98.33	98.82	98.17	**99.81**
6	Stubble	99.41	99.72	99.80	**100**	99.69
7	Celery	99.23	99.46	99.66	97.00	**99.69**
8	Grapes_untrained	61.92	70.37	67.56	70.79	**84.34**
9	Soil_vinyard_develop	98.70	98.59	99.19	**99.45**	98.39
10	Corn_senesced_green_weeds	85.56	93.74	93.80	**96.19**	95.14
11	Lettuce_romaine_4wk	91.75	94.70	95.87	96.37	**98.82**
12	Lettuce_romaine_5wk	98.24	99.89	99.95	**100**	99.63
13	Lettuce_romaine_6wk	97.69	97.81	98.15	**100**	**100**
14	Lettuce_romaine_7wk	92.25	97.35	96.86	98.33	**99.90**
15	Vinyard_untrained	70.32	71.53	80.77	**91.22**	79.36
16	Vinyard_vertical_trellis	96.98	**98.18**	**98.18**	93.00	96.93
OA	-	86.02	88.82	89.72	90.25	**93.01**
AA	-	92.74	94.67	95.43	93.69	**96.56**
Kappa	-	0.8450	0.8757	0.8858	0.8918	**0.9221**

and spectral-spatial classification methods on the four data sets. For spectral classification approaches, CCF-200 outperforms RF-200 and SVM-RBF. With respect to the obtained classification results, deep networks, including SICNN, 2-D CNN, and the proposed SpecAttenNet show better performance than “shallow” models (i.e., random forest, SVM,

TABLE IX

ACCURACY COMPARISONS FOR THE HOUSTON SCENE. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	2D CNN	SpecAttenNet
1	Healthy grass	82.53	82.24	**83.10**	82.91	78.63
2	Stressed grass	83.46	83.18	83.46	**100**	85.81
3	Synthetic grass	97.82	99.80	**100**	75.05	94.65
4	Trees	91.38	92.05	91.48	89.49	**97.82**
5	Soil	96.59	98.58	98.67	99.53	**100**
6	Water	98.60	**99.30**	**99.30**	93.71	89.51
7	Residential	74.81	78.92	**87.59**	76.12	79.10
8	Commercial	32.48	48.81	46.34	71.32	**78.92**
9	Road	69.41	77.90	73.84	80.64	**87.72**
10	Highway	43.73	62.07	66.41	53.96	**70.37**
11	Railway	69.83	81.31	**84.63**	76.57	74.67
12	Parking Lot 1	53.70	81.75	85.98	**88.86**	76.08
13	Parking Lot 2	61.40	71.23	73.68	85.96	**90.53**
14	Tennis Court	99.19	**100**	98.79	81.38	98.38
15	Running Track	97.89	97.04	**98.10**	68.50	95.77
OA	-	72.86	80.80	82.15	81.40	**84.50**
AA	-	76.85	83.61	84.76	81.60	**86.53**
Kappa	-	0.7085	0.7933	0.8069	0.7985	**0.8318**

TABLE X

ASSESSMENTS OF THE SIGNIFICANCE OF CLASSIFICATION ACCURACIES OF THE PROPOSED METHOD COMPARED TO OTHER INVESTIGATED APPROACHES FOR THE FOUR DATA SETS.

Data Set	RF-200	SVM-RBF	CCF-200	SICNN	2D CNN
Indian Pines	40.953	35.169	21.278	19.280	19.255
Pavia University	64.010	36.743	24.161	22.904	16.895
Salinas	2.021	25.101	22.943	-	31.336
Houston	27.389	9.720	6.742	-	8.377

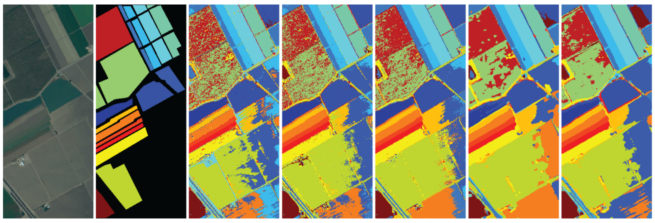


Fig. 5. Classification maps of different approaches for the Salinas data set. From left to right: true-color composite of the hyperspectral image, reference data, RF-200, SVM-RBF, CCF-200, 2-D CNN, and SpecAttenNet. Best zoomed-in view.

and CCF) in regard to OA and kappa coefficient, mainly because: 1) they are capable of extracting hierarchical, deep feature representations; 2) spatial information can be fully exploited in them. These two properties make the deep networks more robust in finding appropriate decision boundaries and enable the models to handle nonlinearly separable data more efficiently.

On the other hand, in comparison with SICNN that selects the most informative spectral bands as inputs of a CNN using a band selection approach, SpecAttenNet is capable of achieving accuracy increments of 7.09%, 2.69%, and 0.0797 for OA, AA, and Kappa coefficient, respectively, on the Indian Pines scene. Regarding the Pavia University scene, the accuracy increments on OA, AA, and Kappa coefficient are, respectively, 3.89%, 2.38%, and 0.0494. This observation reveals that compared to conventional band selection methods, our data- and task-driven spectral attention mechanism can offer better results.

Table X demonstrates the results of McNemar's test, in which we compute our method and other competitors in terms of the significance of the difference between their classification results. We can see that on both data sets, the improvement of accuracies yielded by our approach is statistically significant as compared with other methods. Figs. 3–5 show classification maps produced by RF-200, SVM-RBF, CCF-200, SICNN, 2-D CNN, and SpecAttenNet on three scenes. As displayed in these figures, spectral classifiers (i.e., random forest, SVM, and CCF) lead to salt and pepper noised classification maps, while this issue is addressed in spectral-spatial classification networks (SICNN, 2-D CNN, and SpecAttenNet) by removing noisy scattered points of misclassification.

Moreover, we observe that the use of the spectral attention module alleviates the problem of misclassification. For instance, misclassification in the Indian Pines data set lies in similar objects (with extremely similar spectral characteristics), such as Alfalfa and Hay-windrowed. SpecAttenNet achieves the best average accuracy of 89.625% on these two classes, while the second best average accuracy is only 74.68%, as obtained by SICNN.

E. Analysis of the Spectral Attention Module

One challenge in hyperspectral data classification is that due to complex light scattering mechanism, some pixels of a hyperspectral image, which belong to the same land cover class, have different spectral signatures. Therefore, an approach that is capable of making spectral signals of those pixels that are more similar should be able to offer a more accurate classification result. Here, to quantitatively verify the effectiveness of the spectral attention module, an index called within-class similarity measures is used. The within-class similarity measure is defined as the trace of the

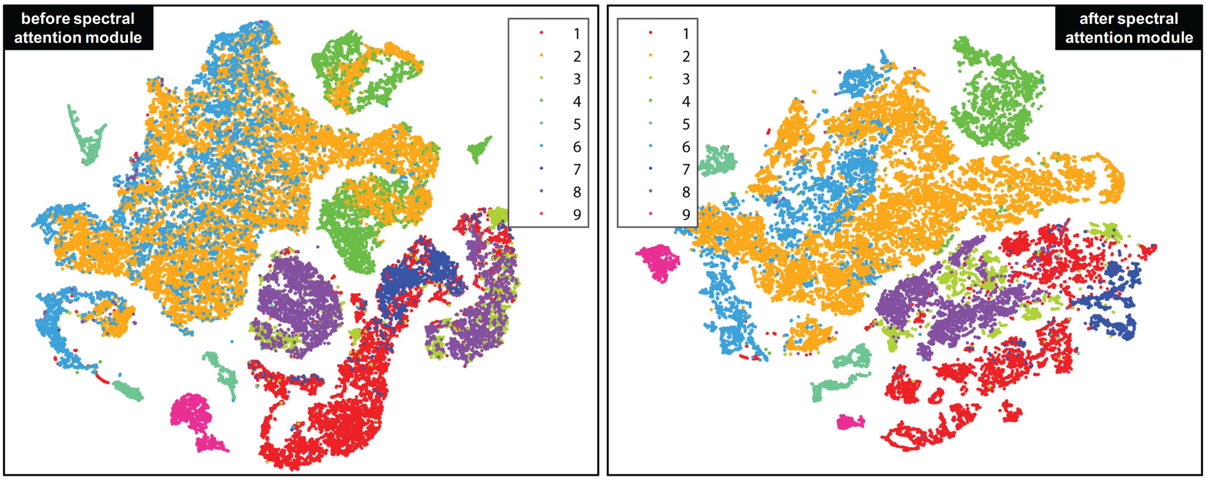


Fig. 6. Visualization of original samples and recalibrated ones by the spectral attention module of the Pavia University data set by t-SNE [70]. Different colors represent different categories. As shown in this figure, after the attention module, samples of some classes (e.g., class 2 and class 6) gather together and come into several groups, which means outputs of the module are more useful for tasks like classification. This is mainly because by making use of the proposed gating mechanism, bands that provide discriminative information are emphasized, while the others are suppressed.

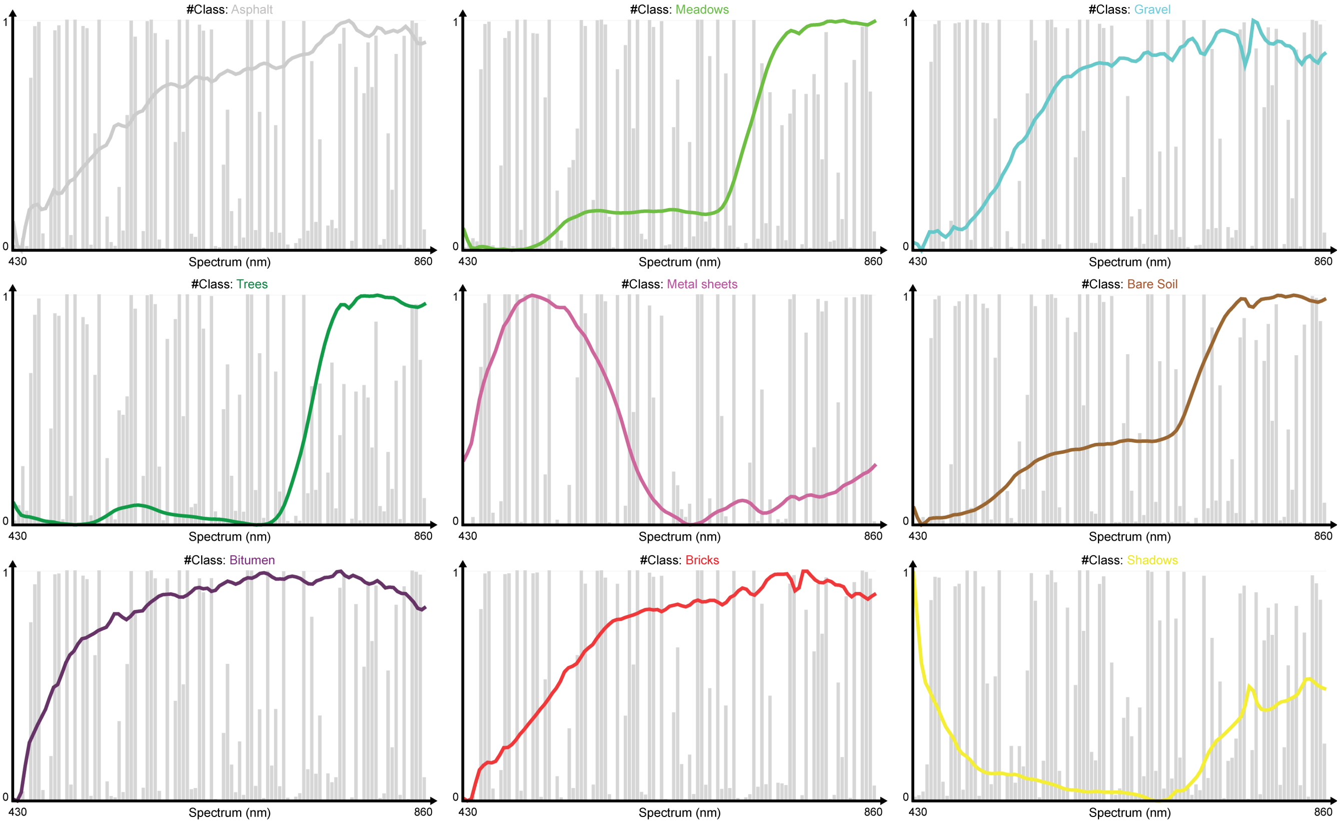


Fig. 7. Average reflectance spectrum and average spectral gates of each class on the Pavia University data set.

within-class scatter matrix, which can be calculated as follows:

$$S_w = \sum_{c} \sum_{i \in c} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T \tag{13}$$

where

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i \in c} \boldsymbol{x}_i \tag{14}$$

and N_c denotes the amount of test data belonging to the c-th category.

Table XI reports calculated within-class similarity measures of features before and after the spectral attention module in our network on both data sets. We can observe that recalibrated spectra (i.e., outputs of the spectral attention module) in the same category have higher similarity.

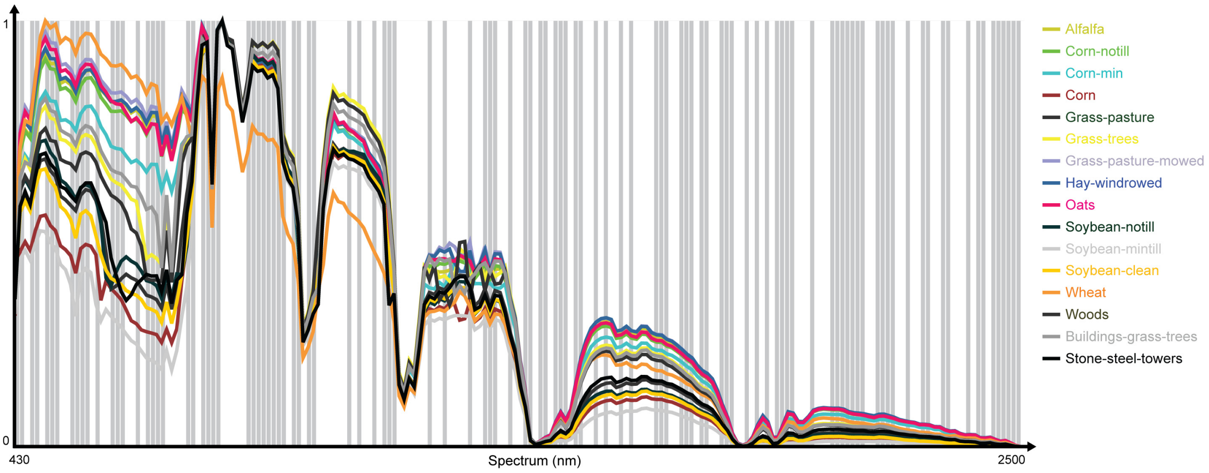


Fig. 8. Average reflectance spectrum of each class and learned spectral gates on the Indian Pines data set.

TABLE XI

WITHIN-CLASS SIMILARITY MEASURES OF FEATURES BEFORE AND AFTER THE SPECTRAL ATTENTION MODULE ON THE INDIAN PINES, PAVIA UNIVERSITY, AND SALINAS DATA SETS. SMALLER IS BETTER

Data Set	Before	After
Indian Pines	17.089	**9.403**
Pavia University	2.289	**1.058**
Salinas	2.240	**0.198**

Hence, the results demonstrate that the recalibrated spectra are more discriminative.

Furthermore, we use t-SNE [70] technique to visualize spectra before and after this module on the Pavia University scene in Fig. 6. As shown in this figure, after the attention module, samples of some classes (e.g., class 2 and class 6) gather together and come into several groups, which means outputs of the module are more useful for tasks like classification. This is mainly because by making use of the proposed gating mechanism, bands that provide discriminative information are emphasized, while others are suppressed.

Since the designed spectral attention mechanism is data- and task-driven, according to (3), different inputs have different spectral gates. For each class, we calculate the average of spectral gates of test samples belonging to this class and name it average spectral gate. Fig. 7 exhibits the average reflectance spectrum and the average spectral gate learned by our attention module of each class on the Pavia University scene. As shown in this figure, classes with similar spectral signatures (e.g., Gravel and Bricks) have extremely similar spectral gates, while these similar classes can be differentiated in detail; for example, we can see that activations of some gates on the Gravel class and the Bricks class are different. In Fig. 8, we also display the average reflectance spectrum of each class and learned spectral gates on the Indian Pines data set. Note that since spectral gates of all test samples learned on this scene are almost the same, we visualize the average spectral gate of all samples instead of each class. Interestingly, the learned spectral gate on this data set is nearly completely binary and quite different from the gates on the Pavia University scene. From Fig. 8, we can observe that the spectral attention module mainly pays attention on spectral bands that provide visual cues to distinguish different categories.

IV. CONCLUSION

This work proposed a simple, yet effective end-to-end trainable spectral attention module to make a spectral-spatial classification CNN learn a channel attention mechanism, i.e., how to pay attention on the spectral domain, for hyperspectral image classification. Our spectral attention module enhances the network by learning the importance of spectral bands with a gating mechanism and performing a dynamic band-wise recalibration, which improves not only the representational capacity but also the interpretability of the network. Extensive experiments validate the effectiveness of our network.

In the future, we will carry out further research and try to figure out the band importance induced by the spectral attention module, which may be helpful to related fields, e.g., band selection and hyperspectral data classification network pruning for model compression.

REFERENCES

[1] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[2] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, “Random forest classifiers for hyperspectral data,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2005, p. 4.

[3] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, “Random Forests for land cover classification,” *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, Mar. 2006.

[4] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[6] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[7] B. Waske and J. A. Benediktsson, "Fusion of support vector machines for classification of multisensor data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3858–3866, Dec. 2007.

[8] B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.

[9] G. F. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2018, pp. 850–860.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3104–3112.

[11] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1746–1751.

[12] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–15.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1–9.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, Apr. 2015, pp. 1–14.

[15] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[16] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[20] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," Jun. 2016, *arXiv:1606.00915*. [Online]. Available: https://arxiv.org/abs/1606.00915

[21] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 91–99.

[23] L. Mou and X. X. Zhu, "Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6699–6711, Nov. 2018.

[24] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," Jun. 2017, *arXiv:1706.06905*. [Online]. Available: https://arxiv.org/abs/1706.06905

[25] L. Mou and X. X. Zhu, "IM2HEIGHT: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network," Feb. 2018, *arXiv:1802.1024*. [Online]. Available: https://arxiv.org/abs/1802.10249

[26] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[27] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.

[28] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[29] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[30] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, 2017.

[31] X. Lu, W. Zhang, and X. Li, "A hybrid sparsity and distance-based discrimination detector for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1704–1717, Mar. 2018.

[32] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[33] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.

[34] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[35] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[36] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.

[37] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.

[38] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep&dense convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 10, no. 9, p. 1454, 2018.

[39] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019. doi: 10.1109/TGRS.2018.2860125.

[40] L. Mou, P. Ghamisi, and X. X. Zhu, "Fully conv-deconv network for unsupervised spectral-spatial feature extraction of hyperspectral imagery via residual learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 5181–5184.

[41] X. Ma, A. Fu, J. Wang, H. Wang, and B. Yin, "Hyperspectral image classification based on deep deconvolution network with skip architecture," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, Aug. 2018.

[42] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2183–2195, Apr. 2018.

[43] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[44] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," Mar. 2018, *arXiv:1803.02642*. [Online]. Available: https://arxiv.org/abs/1803.02642

[45] M. Rußwurm and M. Körner, "Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshop*, Jul. 2017, pp. 1496–1504.

[46] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, p. 506, 2016.

[47] H. Lyu *et al.*, "Long-term annual mapping of four cities on different continents by applying a deep information learning method to Landsat data," *Remote Sens.*, vol. 10, no. 3, p. 471, 2018.

[48] H. Wu and S. Prasad, "Convolutional recurrent neural networks for hyperspectral data classification," *Remote Sens.*, vol. 9, no. 3, p. 298, 2017.

[49] Y. Hua, L. Mou, and X. X. Zhu, "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 149, pp. 188–199, Mar. 2019.

[50] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[51] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 120–147, Nov. 2018.

[52] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–11.

[53] M. E. Paoletti *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[54] F. I. Alam, J. Zhou, A. W.-C. Liew, X. Jia, J. Chanussot, and Y. Gao, "Conditional random field and deep feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1612–1628, Mar. 2019.

[55] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019.

[56] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7190–7198.

[57] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3492–3506, Jul. 2017.

[58] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.

[59] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[60] A. Graves, "Generating sequences with recurrent neural networks," Aug. 2013, *arXiv:1308.0850*. [Online]. Available: https://arxiv.org/abs/1308.0850

[61] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl. (SSST)*, Oct. 2014, pp. 103–167.

[62] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1019–1027.

[63] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published. doi: 10.1109/TGRS.2019.2918080.

[64] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.

[65] T. Dozat. *Incorporating Nesterov Momentum Into Adam*. Accessed: Sep. 22, 2019. [Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf

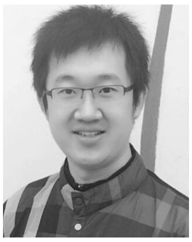
[66] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[67] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[68] T. Rainforth and F. Wood, "Canonical correlation forests," Jul. 2015, *arXiv:1507.05444*. [Online]. Available: https://arxiv.org/abs/1507.05444

[69] J. Xia, N. Yokoya, and A. Iwasaki, "Hyperspectral image classification with canonical correlation forests," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 421–431, Jan. 2017.

[70] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014.



Lichao Mou (S'16) received the bachelor's degree in automation from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2012, and the master's degree in signal and information processing from the University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the German Aerospace Center (DLR), Wessling, Germany, and also with the Technical University of Munich (TUM), Munich, Germany.

In 2015, he spent six months at the Computer Vision Group, University of Freiburg, Freiburg im Breisgau, Germany. In 2019, he was a Visiting Researcher with the University of Cambridge, Cambridge, U.K. His research interests include remote sensing, computer vision, and machine learning, especially deep networks and their applications in remote sensing.

Mr. Mou was a recipient of the first place in the 2016 IEEE GRSS Data Fusion Contest and finalists for the Best Student Paper Award at the 2017 Joint Urban Remote Sensing Event and the 2019 Joint Urban Remote Sensing Event.

Xiao Xiang Zhu (S'10–M'12–SM'14) received the master's (M.Sc.), D.E. (Dr.-Ing.), and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She was a Guest Scientist or a Visiting Professor with the Italian National Research Council (CNR-IREA), Naples, Italy, in 2009; Fudan University, Shanghai, China, in 2014; The University of Tokyo, Tokyo, Japan, in 2015; and the University of California at Los Angeles, Los Angeles, CA, USA, in 2016. Since 2019, she has been co-coordinating the Munich Data Science Research School. She is also leading the Helmholtz Artificial Intelligence Cooperation Unit (HAICU)–Research Field "Aeronautics, Space and Transport." She is currently a Professor of signal processing in earth observation with the Technical University of Munich (TUM) and also with the German Aerospace Center (DLR); also the Head of the Department "EO Data Science," DLR's Earth Observation Center; and also the Head of the Helmholtz Young Investigator Group "SiPEO," DLR and TUM. Her research interests include remote sensing and Earth observation, signal processing, machine learning, and data science, with a special application focus on global urban mapping.

Dr. Zhu is a member of young academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities, the German National Academy of Sciences Leopoldina, and the Bavarian Academy of Sciences and Humanities. She is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

E Mou L., Zhu X., 2018. Vehicle Instance Segmentation from Aerial Image and Video Using a Multi-Task Learning Residual Fully Convolutional Network, IEEE Transactions on Geoscience and Remote Sensing, 56(11), 6699-6711.

Vehicle Instance Segmentation From Aerial Image and Video Using a Multitask Learning Residual Fully Convolutional Network

Lichao Mou, *Student Member, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*— Object detection and semantic segmentation are two main themes in object retrieval from high-resolution remote sensing images, which have recently achieved remarkable performance by surfing the wave of deep learning and, more notably, convolutional neural networks. In this paper, we are interested in a novel, more challenging problem of vehicle instance segmentation, which entails identifying, at a pixel level, where the vehicles appear as well as associating each pixel with a physical instance of a vehicle. In contrast, vehicle detection and semantic segmentation each only concern one of the two. We propose to tackle this problem with a semantic boundary-aware multitask learning network. More specifically, we utilize the philosophy of residual learning to construct a fully convolutional network that is capable of harnessing multilevel contextual feature representations learned from different residual blocks. We theoretically analyze and discuss why residual networks can produce better probability maps for pixelwise segmentation tasks. Then, based on this network architecture, we propose a unified multitask learning network that can simultaneously learn two complementary tasks, namely, segmenting vehicle regions and detecting semantic boundaries. The latter subproblem is helpful for differentiating "touching" vehicles that are usually not correctly separated into instances. Currently, data sets with a pixelwise annotation for vehicle extraction are the ISPRS data set and the IEEE GRSS DFC2015 data set over Zeebrugge, which specializes in a semantic segmentation. Therefore, we built a new, more challenging data set for vehicle instance segmentation, called the *Busy Parking Lot Unmanned Aerial Vehicle Video data set*, and we make our data set available at http://www.sipeo.bgu.tum.de/downloads so that it can be used to benchmark future vehicle instance segmentation algorithms.**

***Index Terms*— Boundary-aware multitask learning network, fully convolutional network (FCN), high-resolution remote sensing image/video, instance semantic segmentation, residual neural network (ResNet), vehicle detection.**

I. Introduction

The last decade has witnessed dramatic progress in modern remote sensing technologies—along with the launch of small and cheap commercial high-resolution satellites and the now widespread availability of unmanned aerial vehicles (UAVs)—which facilitates a diversity of applications, such as urban management [1]–[4], monitoring of land changes [5]–[8], and traffic monitoring [9], [10]. Among these applications, object extraction from very high-resolution remote sensing images/videos has gained increasing attention in the remote sensing community in recent years, particularly vehicle extraction, due to successful civil applications. Vehicle extraction, however, is still a challenging task, mainly because it is easily affected by several factors, e.g., vehicle appearance variation, the effects of shadow, illumination, and a complicated and cluttered background. Existing vehicle extractionapproaches can be roughly divided into two categories: vehicle detection and vehicle semantic segmentation.

A. Vehicle Detection

The goal of vehicle detection is to detect all instances of vehicles and localize them in the image, typically in the form of bounding boxes with confidence scores. Traditionally, this topic was addressed by works that use low-level, hand-crafted visual features [e.g., color histogram, texture feature, scale-invariant feature transform (SIFT), and histogram of oriented gradients (HOG)] and classifiers. For example, Shao *et al.* [11] incorporate multiple visual features, local binary patterns, HOG, and opponent histogram for vehicle detection from high-resolution aerial images. Moranduzzo and Melgani [12] first use SIFT to detect the interest points of vehicles and then train a support vector machine (SVM) to classify these interest points into vehicle and nonvehicle categories based on the SIFT descriptors. They later present an approach [13] that performs filtering operations in the horizontal and vertical directions to extract HOG features and yield vehicle detection after the computation of a similarity measure, using a catalog of vehicles as a reference. Liu and Mattyus [14] make use of an integral channel concept with Haar-like features and an AdaBoost classifier in a soft-cascade structure to achieve fast and robust vehicle detection.

The aforementioned approaches mainly rely on the hand-crafted features for constructing a classification system.

Manuscript received November 18, 2017; revised April 2, 2018 and May 22, 2018; accepted May 23, 2018. Date of publication July 9, 2018; date of current version October 25, 2018. This work was supported in part by the China Scholarship Council, in part by the European Research Council through the European Union's Horizon 2020 Research and Innovation Programme under Grant ERC-2016-StG-714087 (So2Sat), in part by the Helmholtz Association through the Framework of the Young Investigators Group (SiPEO) under Grant VH-NG-1018, and in part by the Bavarian Academy of Sciences and Humanities through the Framework of Junges Kolleg. *(Corresponding author: Xiao Xiang Zhu.)*

The authors are with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Wessling, Germany, and also with Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: lichao.mou@dlr.de; xiao.zhu@dlr.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2841808

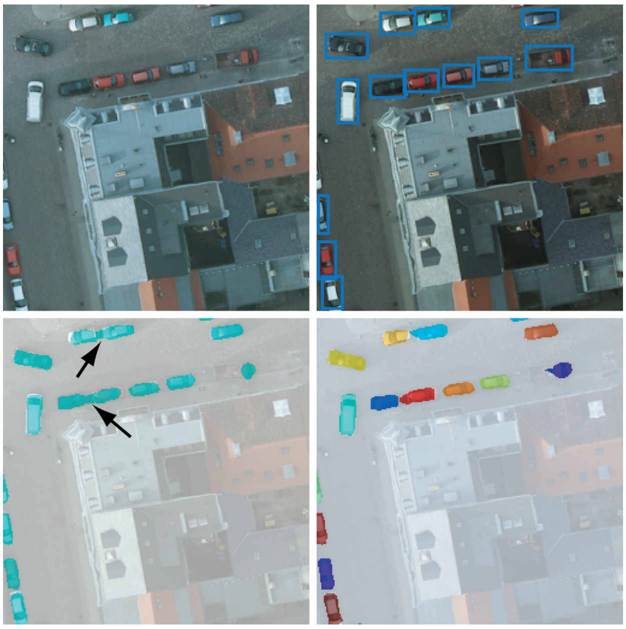


Fig. 1. Illustration of different vehicle extraction methods. (From left to right and top to bottom) Input image, vehicle detection, semantic segmentation, and vehicle instance segmentation. The challenge of vehicle instance segmentation is that some vehicles are segmented incorrectly. While most pixels belonging to the category are identified correctly, they are not correctly separated into instances (see arrows in the bottom-left image).

Recently, as an important branch of the deep learning family, the convolutional neural network (CNN) has become the method of choice in many computer vision and remote sensing problems [15]–[19] (e.g., object detection) due to its ability to automatically extract midlevel and high-level abstract features from raw images for pattern recognition purposes. Chen *et al.* [20] propose a vehicle detection model, called the hybrid deep neural network, which consists of a sliding window technique and CNN. The main insight behind their model is to divide the feature maps of the last convolutional layer into different scales, allowing for the extraction of multiscale features for vehicle detection. Ammour *et al.* [21] segment an input image into homogeneous superpixels that can be considered as vehicle candidate regions, making use of a pretrained deep CNN to extract features, and train a linear SVM to classify these candidate regions into vehicle and nonvehicle classes.

B. Vehicle Semantic Segmentation

Vehicle semantic segmentation aims to label each pixel in an image as belonging to the vehicle class or other categories (e.g., building, tree, and low vegetation). In comparison with vehicle detection, it can give more accurate pixelwise extraction results. More recently, progress in deep CNNs, particularly fully convolutional networks (FCNs), makes it possible to achieve end-to-end vehicle semantic segmentation. For instance, Audebert *et al.* [22] propose a deep-learning-based "segment-before-detect" method for semantic segmentation and subsequent classification of several types of vehicles in high-resolution remote sensing images. The use of SegNet [23] in this method is capable of producing pixelwise annotations for vehicle semantic mapping. In addition, several recent works in the semantic segmentation of high-resolution aerial imaging also involve vehicle segmentation. Kampffmeyer *et al.* [24] focus on the class imbalance which often represents a problem for semantic segmentation in remote sensing images, since small objects (e.g., vehicles) are less prioritized in an effort to achieve a good overall accuracy (OA). To address this problem, they train FCNs using the cross-entropy loss function weighted with median frequency balancing, which is proposed by Eigen and Fergus [25].

C. Is Semantic Segmentation Good Enough for Vehicle Extraction?

The existence of "touching" vehicles in a remote sensing image makes it quite hard for most vehicle semantic segmentation methods to separate objects individually, while in most cases, we need to know not only which pixels belong to vehicles (vehicle semantic segmentation problem) but also the exact number of vehicles (vehicle detection task). This drives us to examine an instance-oriented vehicle segmentation.

The vehicle instance segmentation seeks to identify the semantic class of each pixel (i.e., vehicle or nonvehicle) as well as associate each pixel with a physical instance of a vehicle. This is contrasted with the vehicle semantic segmentation which is only concerned with the above-mentioned first task. Fig. 1 shows differences among vehicle detection, semantic segmentation, and instance segmentation. In this paper, we are interested in the vehicle instance segmentation in a complex, cluttered, and challenging background from aerial images and videos. Moreover, since deep networks have recently been very successful in a variety of remote sensing applications, from hyperspectral/multispectral image analysis to interpretation of high-resolution aerial images to multimodal data fusion [15], in this paper, we would like to use an end-to-end network to achieve the vehicle instance segmentation. This paper contributes to the literature in three major respects.

1) So far, most studies in the remote sensing community have focused on the object detection and semantic segmentation in high-resolution remote sensing imagery. The instance segmentation has rarely been addressed. In a pioneer work moving from semantic segmentation to instance segmentation, Audebert *et al.* [22] developed a three-stage segment-before-detect framework. In this paper, we try to address the vehicle instance segmentation problem by an end-to-end learning framework.
2) In order to facilitate progress in the field of vehicle instance segmentation in high-resolution aerial images/videos, we provide a new, challenging data set that presents a high range of variation—with a diversity of vehicle appearances, the effects of shadow, a cluttered background, and extremely close vehicle distances—for producing quantitative measurements and comparing among approaches.
3) We present a semantic boundary-aware unified multitask learning FCN, which is end-to-end trainable, for vehicle instance segmentation. Inspired by several recent works [26]–[28], we exploit residual neural network (ResNet) [29] to construct the feature extractor

of the whole network. In this paper, we theoretically analyze and discuss why residual networks can produce better probability maps for pixelwise prediction tasks. The proposed multitask learning network creates two separate, yet identical branches to jointly optimize two complementary tasks—namely, vehicle semantic segmentation and semantic boundary detection. The latter subproblem is beneficial for differentiating vehicles with an extremely close distance and further improving the instance segmentation performance.

The remainder of this paper is organized as follows. After Section I, detailing vehicle extraction from high-resolution remote sensing imagery, we enter Section II, dedicated to the details of the proposed semantic boundary-aware multitask learning network for vehicle instance segmentation. Section III then provides the data set information, the network setup, and the experimental results and discussion. Finally, Section IV concludes this paper.

II. Methodology

We formulate the vehicle instance segmentation task by two subproblems, namely, vehicle detection and semantic segmentation. The training set is denoted by $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)\}$, where $i = 1, 2, \ldots, N$ and N is the number of training samples. Since we consider each image independently, the subscript i is dropped hereafter for notational simplicity. $\boldsymbol{x} = \{x_j, j = 1, 2, \ldots, |\boldsymbol{x}|\}$ represents a raw input image, $\boldsymbol{y} = \{y_j, j = 1, 2, \ldots, |\boldsymbol{x}|, y_j \in \{0, 1\}\}$ denotes its corresponding manually annotated pixelwise segmentation mask, and $\boldsymbol{z} = \{\boldsymbol{r}_k, k = 0, 1, \ldots, K\}$ is the instance label, where $\boldsymbol{r}_k$ indicates a set of pixels inside the kth region.[1] K is the total number of vehicle instances in the image, and $\boldsymbol{r}_0$ is the background area. When k takes other values, it denotes the corresponding vehicle instance. Note that the instance labels only count vehicle instances, and thus, they are commutative. Our aim is to segment vehicles while ensuring that all instances are differentiated. In this paper, we approximate the vehicle detection by the semantic boundary detection.[2] We generate the semantic boundary labels $\boldsymbol{b}$ through $\boldsymbol{z}$ to train a boundary detector, in which $\boldsymbol{b} = \{b_j, j = 1, 2, \ldots, |\boldsymbol{x}|, b_j \in \{0, 1\}\}$ and b_j equals 1 when it belongs to boundaries.

In this section, we describe our proposed semantic boundary-aware multitask learning network for accurate vehicle instance segmentation in detail. We start by introducing the FCN architecture for end-to-end semantic segmentation in Section II-A. Furthermore, we propose to exploit multilevel contextual feature representations, generated by different stages of a residual network, to construct a residual FCN (ResFCN) for producing better likelihood maps of vehicle regions or semantic boundaries (see Section II-B). Then, in Section II-C, we elaborate the semantic boundary-aware unified multitask learning network drawn from the ResFCN for effective instance segmentation by jointly optimizing the complementary tasks.

A. Fully Convolutional Network for Semantic Segmentation

Long *et al.* [30] first proposed the FCN architecture for semantic segmentation tasks which is both efficient and effective. Later, some extensions of the FCN model have been proposed to improve a semantic segmentation performance. To name a few, Chen *et al.* [31] removed some of the max-pooling operations and, accordingly, introduced atrous/dilated convolutions in their network, which can expand the field of view without increasing the number of parameters. As postprocessing, a dense conditional random field (CRF) was trained separately to refine the estimated category score maps for further improvement. Zhang *et al.* [32] introduced a new form of network that combines FCN- and CRF-based probabilistic graphical modeling to simulate a mean-field approximate inference for the CRF with Gaussian pairwise potentials as the recurrent neural network.

B. Residual Fully Convolutional Network

Here, we first explain how to construct a ResFCN according to the existing works in the literature, mainly, the ResNet [29] and FCN [30]. Then, we theoretically analyze why ResFCN is able to offer better performance than other FCNs based on the traditional feedforward network architectures (e.g., VGG Nets [33]).

Network Design: Several recent studies in computer vision [26]–[28] have shown that ResNet [29] is capable of offering better features for pixelwise prediction tasks, such as semantic segmentation [26], [27] and depth estimation [28]. We, therefore, make use of ResNet to construct the segmentation network in this paper. We initialize a ResFCN from the original version of ResNet [29], instead of the newly presented preactivation version [34]. Unlike [30], we directly remove the fully connected layers from the original ResNet but do not convolutionalize these layers so as to make one prediction per spatial location. Moreover, we keep the 7 × 7 convolutional layer and 3 × 3 max-pooling layer, which can enlarge the field of view for feature representations. One of the recent trends in a network architecture design is stacking convolutional layers with small convolution kernels (e.g., 3×3 and 1×1) in the entire network, because the stacked small kernels are more efficient than a large filter, given the same computational complexity. However, a recent study [35] found that the large filter also plays an important role when classification and localization tasks are performed simultaneously. This can be easily understood through the analogy of individuals commonly confirming the category of a pixel by referring to its surrounding context region.

By now, the output feature maps are only 1/32 the resolution of their original input image, which is apparently too low to precisely differentiate individual pixels. To deal with this problem, Long *et al.* [30] made use of backward-strided convolutions that upsample the feature maps and output score masks. The motivation behind this is that the convolutional layers and max-pooling layers focus on extracting high-level abstract features, whereas the backward-strided convolutions estimate the score masks in a pixelwise way. Ghiasi and Fowlkes [36] proposed a multiresolution recon-

[1] Regions in the image satisfy $\boldsymbol{r}_k \cap \boldsymbol{r}_t = \varnothing, \forall k \neq t$ and $\cup \boldsymbol{r}_k = \Omega$, where Ω is the whole image region.

[2] The semantic boundary detection is to detect the boundaries of each object instance in the images. Compared with edge detection, it focuses more on the association of boundaries and their object instances.

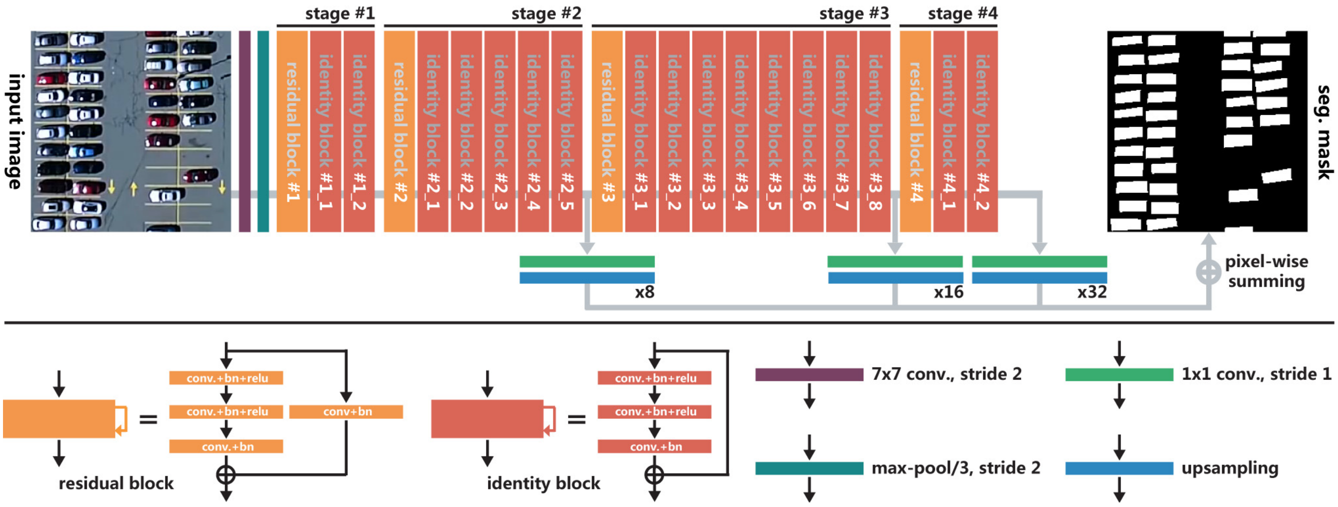


Fig. 2. Network architecture of the ResFCN we use, as illustrated in Section II-B. We incorporate the multilevel contextual features from the last 32 × 32, 16 × 16, and 8 × 8 layers of a classification ResNet since making use of information from fairly early fine-grained layers is beneficial to segmenting small objects such as vehicles. To get the desired full resolution output, we use 1 × 1 convolutional layers followed by upsampling operations to upsample back to the spatial resolution of the input image. Then, predictions from different residual blocks are fused together with a summing operation.

struction architecture based on a Laplacian pyramid that uses skip connections from higher resolution feature maps and multiplicative gating to successively refine segment boundaries reconstructed from lower resolution maps. Inspired by the existing works, in this paper, we exploit multilevel contextual feature representations that include information from different residual blocks (i.e., different levels of contextual information). Fig. 2 shows the illustration of the ResFCN architecture we use with multilevel contextual features. More specifically, we incorporate feature representations from the last 32 × 32, 16 × 16, and 8 × 8 layers of the original ResNet, since making use of information from fairly early fine-grained layers is beneficial to segmenting small objects such as vehicles. To get the desired full resolution output, we used a 1 × 1 convolutional layer, which adaptively squashes the number of channels down to the number of labels (1 for binary classification), takes advantage of the upsampling operation to upsample back to the spatial resolution of the input image, and makes predictions based on the contextual cues from the given fields of view. Then, these predictions are fused together with a summing operation, and the final segmentation results are generated after sigmoid classification.

Why Residual Learning? Until recently, the majority of feedforward networks, such as AlexNet [37] and VGG Nets [33], were made up of a linear sequence of layers. $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_n$ are denoted as the input and output of the nth layer/block, respectively, and each layer in such a network learns the mapping function $\mathcal{F}$

$$\boldsymbol{x}_n = \mathcal{F}(\boldsymbol{x}_{n-1}; \boldsymbol{\Theta}_n) \tag{1}$$

where $\boldsymbol{\Theta}_n$ is the parameters of the nth layer. This kind of network is also often referred to as a traditional feedforward network.

According to a study by He *et al.* [29], simply deepening traditional feedforward networks usually leads to an increase in training and test errors (i.e., so-called degradation problem). A residual learning-based network is composed of a sequence of residual blocks and exhibits significantly improved training characteristics, providing the opportunity to make network depths that were previously unattainable. The output $\boldsymbol{x}_n$ of the nth residual block in a ResNet can be computed as

$$\boldsymbol{x}_n = \mathcal{H}(\boldsymbol{x}_{n-1}; \boldsymbol{\Theta}_n) + \boldsymbol{x}_{n-1} \tag{2}$$

where $\mathcal{H}(\boldsymbol{x}_{n-1}; \boldsymbol{\Theta}_n)$ is the residual, which is parametrized by $\boldsymbol{\Theta}_n$. The core insight of ResNet is that the addition of a shortcut connection from the input $\boldsymbol{x}_{n-1}$ to the output $\boldsymbol{x}_n$ bypasses two or more convolutional layers by performing identity mapping and is then added together with the output of stacked convolutions. By doing so, $\mathcal{H}$ only computes a residual instead of computing the output $\boldsymbol{x}_n$ directly.

In the experiments, we found that the ResFCN can offer a better performance than the other FCNs based on the traditional feedforward network architecture, such as VGG-FCN. What is the reason behind this? To answer this question, we need to go deeper. According to the characteristics of the ResFCN, we can easily get the following recurrence formula:

$$\boldsymbol{x}_m = \sum_{i=n-1}^{m-1} \mathcal{H}(\boldsymbol{x}_i; \boldsymbol{\Theta}_{i+1}) + \boldsymbol{x}_{n-1} \tag{3}$$

for any deeper residual block m and any shallower residual block n. Equation (3) shows that the ResFCN creates a direct path for propagating information of shallow layers (i.e., $\boldsymbol{x}_{n-1}$) through the entire network. Several recent studies [38], [39] that attempt to reveal what were learned by CNNs show that the deeper layers exploit filters to grasp global high-level information, while the shallower layers capture low-level details, such as object boundaries and edges, which are of great importance in small object detection/segmentation. In addition, when we dive into the backward propagation process,

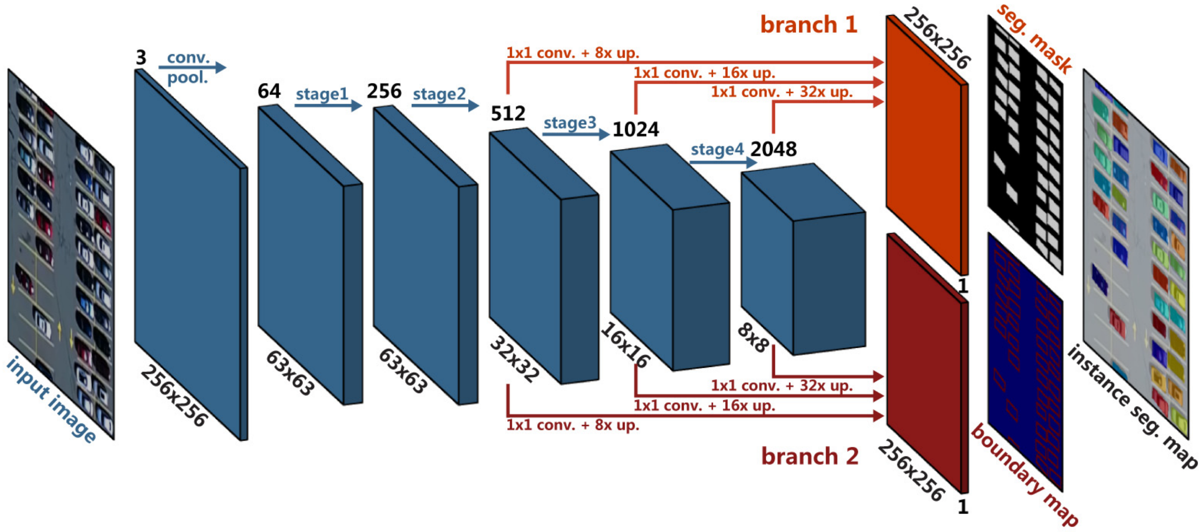


Fig. 3. Overall architecture of the proposed semantic B-ResFCN. We propose to use such a unified multitask learning network for vehicle instance segmentation which creates two separate, yet identical branches to jointly optimize two complementary tasks, namely, vehicle semantic segmentation and semantic boundary detection. The latter subproblem is beneficial for differentiating "touching" vehicles and further improving the instance segmentation performance.

according to the chain rule of backpropagation, we can obtain

$$\begin{aligned}\frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_{n-1}} &= \frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_m}\frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_{n-1}} \\ &= \frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_m}\left(1+\frac{\partial}{\partial \boldsymbol{x}_{n-1}}\sum_{i=n-1}^{m-1}\mathcal{H}(\boldsymbol{x}_i;\boldsymbol{\Theta}_{i+1})\right) \qquad (4)\end{aligned}$$

where $\mathcal{E}$ is the loss function of the network. As exhibited in (4), the gradient $(\partial\mathcal{E}/(\partial\boldsymbol{x}_{n-1}))$ can be decomposed into two additive terms: the term $(\partial\mathcal{E}/(\partial\boldsymbol{x}_m))((\partial/(\partial\boldsymbol{x}_{n-1}))\sum_{i=n-1}^{m-1}\mathcal{H})$ that passes information through the weight layers, and the term $(\partial\mathcal{E}/(\partial\boldsymbol{x}_m))$ that directly propagates without concerning any weight layers. The latter term ensures that the information can also be directly propagated back to any shallower residual block n.

In brief, the properties of the forward and backward propagation procedures of the ResFCN make it possible to shuttle the low-level visual information directly across the network, which is quite helpful for our vehicle (small object) instance segmentation tasks.

C. Semantic Boundary-Aware ResFCN

By exploiting the multilevel contextual features, the ResFCN is capable of producing good likelihood maps of vehicles. However, it is still difficult to differentiate vehicles with a very close distance by only leveraging the probability of vehicles due to the ambiguity in the "touching" regions. This is rooted in the loss of spatial details caused by max-pooling layers (downsampling) along with the feature abstraction. The semantic boundaries of vehicles provide good complementary cues that can be used for separating the instances.

Some approaches in computer vision and remote sensing have been explored for modeling segmentation and boundary prediction jointly in a combinatorial framework. For example, Kirillov *et al.* [40] propose InstanceCut which represents instance segmentation by two modalities, namely, a semantic segmentation and all instance boundaries. The former is computed from a CNN for semantic segmentation, and the latter is derived from an instance-aware edge detector. However, this approach does not address end-to-end learning. In the remote sensing community, Marmanis *et al.* [41] propose a two-step model that learns a CNN to separately output edge likelihoods at multiple scales from color-infrared and height data. Then, the boundaries detected with each source are added as an extra channel to each source, and a network is trained for semantic segmentation purposes. The intuition behind this paper is that using predicted boundaries helps to achieve sharper segmentation maps. In contrast, we train one end-to-end network that takes as input color images and predicts segmentation maps and object boundaries in order to augment the performance of segmentation at the instance level.

To this end, we train a deep semantic boundary-aware ResFCN (B-ResFCN) for effective vehicle instance segmentation (i.e., segmenting the vehicles and splitting clustered instances into individual ones). Fig. 3 shows an overview of the proposed network. Specifically, we formulate it as a unified multitask learning network architecture by exploring the complementary information (i.e., vehicle region and semantic boundaries), instead of treating the vehicle segmentation problem as an independent and single task, which can simultaneously learn the detections of vehicle regions and corresponding semantic boundaries. As shown in Fig. 3, the feature representations extracted from multiple residual blocks are upsampled with two separate, yet identical branches to predict the semantic segmentation masks of vehicles and semantic boundaries, respectively. In each branch, the mask is estimated by the ResFCN with multilevel contextual features, as illustrated in Section II-B. Since we have only two categories (foreground/vehicles versus background and semantic boundaries versus nonboundaries), sigmoid and binary cross-entropy loss are used to train these two branches. Formally, the network training can be formulated as a pixel-level binary classification problem regarding ground-truth segmentation masks, including vehicle instances and semantic boundaries, as shown in the following:

$$\mathcal{L}(x; \boldsymbol{W}) = \mathcal{L}_s(x; \boldsymbol{W}_n, \boldsymbol{W}_s) + \lambda \mathcal{L}_b(x; \boldsymbol{W}_n, \boldsymbol{W}_b) \tag{5}$$

where

$$\begin{aligned}\mathcal{L}_s &= -\sum_{x \in \boldsymbol{x}} [y \log \sigma_s(x) + (1-y)\log(1-\sigma_s(x))] \\ \mathcal{L}_b &= -\sum_{x \in \boldsymbol{x}} [b \log \sigma_b(x) + (1-b)\log(1-\sigma_b(x))].\end{aligned} \tag{6}$$

$\mathcal{L}_s(x; \boldsymbol{W}_n, \boldsymbol{W}_s)$ and $\mathcal{L}_b(x; \boldsymbol{W}_n, \boldsymbol{W}_s)$ denote the losses for estimating vehicle regions and semantic boundaries, respectively. We train the network using this joint loss, and the final instance segmentation map is produced by the first branch of the network in the test phase. Vehicle instances are obtained by computing the connected regions in the predicted segmentation map. Inside a region, pixels belong to the same vehicle, while different regions mean different instances. Our motivation is that jointly estimating segmentation and boundary map in a multitask network with such a joint loss can offer a better segmentation result at the instance level for aerial images. Note that we do not make use of any postprocessing operations, such as fusing the segmentation and boundary map, as we want to directly evaluate the performance of this network architecture.

Note that the multitask learning network is optimized in an end-to-end fashion. This joint multitask training procedure has several merits. First, in the application of vehicle instance segmentation, the multitask learning network architecture is able to provide the complementary semantic boundary information, which is helpful in differentiating the clustered vehicles, improving the instance-level segmentation performance. Second, the discriminative capability of the network's intermediate feature representations can be improved by this architecture because of multiple regularizations on correlated tasks. Therefore, it can increase the robustness of instance segmentation performance.

III. Experimental Results and Discussion

A. Data Sets

1) ISPRS Potsdam: The ISPRS Potsdam Semantic Labeling data set [42] is an open benchmark data set provided online.[3] The data set consists of 38 orthorectified aerial IRRGB images (6000×6000 pixels) with a 5-cm spatial resolution and the corresponding DSMs generated by dense image matching, taken over the city of Potsdam, Germany. A comprehensive manually annotated pixelwise segmentation mask is provided as the ground truth for 24 tiles that are available for training and validation. The other 14 remain unreleased and are kept with the challenge organizers for testing purposes. We randomly selected five tiles (image number: 2_12, 5_12, 7_7, 7_8, 7_9) from 24 training images and used them as the test set in our experiments (see Fig. 4). The resolution is downsampled to 15 cm/pixels to match the subsequent video data set. The input to the networks contains only red, green, and blue channels, and all the results reported on this data set refer to the aforementioned test set. Table I provides the details about this data set for our experiments.

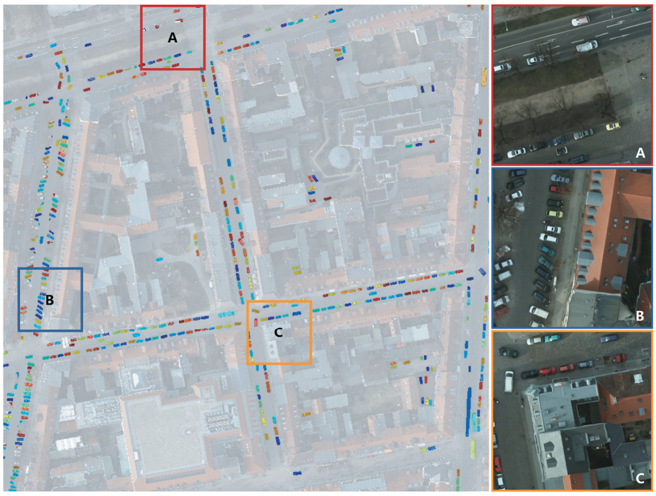


Fig. 4. Image #5_12 from the ISPRS Potsdam data set for vehicle instance segmentation as well as three zoomed-in areas.

2) Busy Parking Lot: The task of vehicle instance segmentation currently lacks a compelling and challenging benchmark data set to produce quantitative measurements and to compare with other approaches. While the ISPRS Potsdam data set has clearly boosted research in semantic segmentation of high-resolution aerial imagery, it is not as challenging as certain practical scenes, such as a busy parking lot, where vehicles are often parked so close that it is quite hard to separate them, particularly from an aerial view. To this end, in this paper, we propose our new challenging Busy Parking Lot UAV Video data set that we built for the vehicle instance segmentation task. The UAV video was acquired by a camera onboard, a UAV covering the parking lot of Woburn Mall, Woburn, MA, USA.[4] The video comprises 1920 × 1080 pixels with a spatial resolution of about 15 cm per pixel at 24 frames/s and a length of 60 s. We have manually annotated pixelwise instance segmentation masks for 5 frames (at 1, 15, 30, 45, and 59 s), i.e., the annotation is dense in space and sparse in time to allow for the evaluation of methods with this long sequence (see Fig. 6). The Busy Parking Lot data set is challenging because it presents a high range of variations with a diversity of vehicle colors, the effects of shadow, several slightly blurred regions, and vehicles that are parked too close. We train the networks on the ISPRS Potsdam data set and then perform vehicle instance segmentation using the trained networks on this video data set. Details regarding this data set are shown in Table II.

B. Training Details

The network training is based on the TensorFlow framework. We choose Nesterov Adam [43], [44] as the optimizer to train the network, since for this task, it shows much faster

[3] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html

[4] https://www.youtube.com/watch?v=yojapmOkIfg

TABLE I
VEHICLE COUNTS AND NUMBER OF VEHICLE PIXELS IN THE ISPRS POTSDAM DATA SET

	Training Set	Test Set				
		2_12	5_12	7_7	7_8	7_9
Vehicle Count	4,433	123	427	301	309	305
Number of Pixels	1,184,789	36,236	122,332	76,892	77,669	74,404

TABLE II
VEHICLE COUNTS AND NUMBER OF VEHICLE PIXELS IN THE BUSY PARKING LOT UAV VIDEO DATA SET

	Frame@1s	Frame@15s	Frame@30s	Frame@45s	Frame@59s
Vehicle Count	511	492	502	484	479
Number of Pixels	257,462	235,560	240,607	235,448	226,697

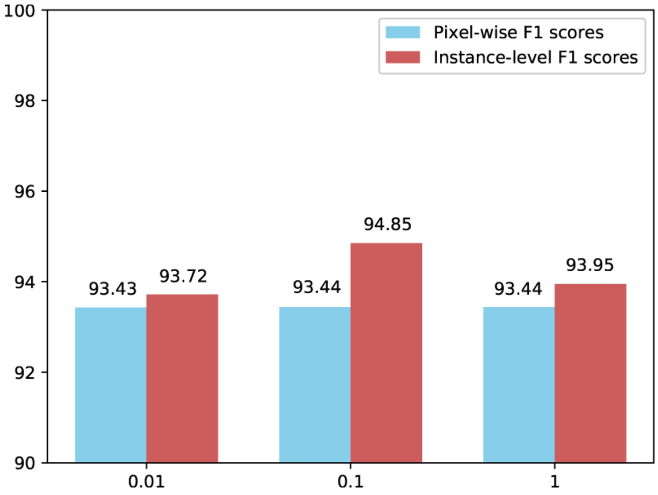


Fig. 5. Sensitivity analysis for the parameter λ on the ISPRS Potsdam data set.

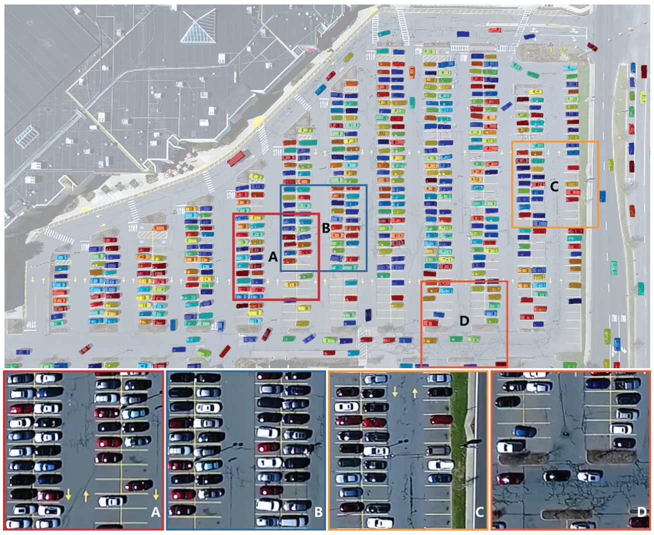


Fig. 6. Frame@1s from the proposed Busy Parking Lot UAV Video data set for vehicle instance segmentation. (Bottom) Four zoomed-in areas.

convergence than the standard stochastic gradient descent with momentum [45] or Adam [46]. We fixed almost all of the parameters of Nesterov Aadam as recommended in [43]: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e{-}08$, and a schedule decay of 0.004, making use of a fairly small learning rate of 2e−04. All weights in the newly added layers are initialized with a Glorot uniform initializer [47] that draws samples from a uniform distribution. In our experiments, we note that the pixelwise F1 score of the network is less sensitive to the parameter λ and the instance-level performance is relatively sensitive to λ. Based on the sensitivity analysis (see Fig. 5), we set it as 0.1.

The networks are trained on the training set of the ISPRS Potsdam data set to predict instance segmentation maps. The training set has only 931 unique 256 × 256 patches. We make use of the data augmentation technique to increase the number of training samples. The RGB patches and the corresponding pixelwise ground truth are transformed by horizontally and vertically flipping three quarters of the patches. By doing so, the number of training samples increases to 14 896. To monitor overfitting during training, we randomly select 10% of the training samples as the validation set, i.e., splitting the training set into 13 406 training and 1490 validation pairs. We train the network for 50 epochs and make use of early stopping to avoid overfitting. Moreover, we use fairly small mini-batches of eight image pairs because, in a sense, every pixel is a training sample. We train our network on a single NVIDIA GeForce GTX TITAN with 12 GB of GPU memory, which takes about 2 h.

C. Qualitative Evaluation

Some vehicle instance segmentation results are shown in Fig. 7 (test set of the ISPRS Potsdam data set) and Fig. 9 (the Busy Parking Lot data set), respectively, in order to qualitatively illustrate the efficacy of our model. First, we compare various CNN variants used for FCN architecture to determine which one is the best suited for our task. In Fig. 7, we qualitatively investigate the accuracy of the predicted instance segmentation maps using FCN architecture with leading CNN variants, namely, VGG-FCN [33], Inception-FCN [48], Xception-FCN [49], and ResFCN on the ISPRS Potsdam data set. We implement VGG-FCN, Inception-FCN, and Xception-FCN by fusing the output feature maps of the last three convolutional blocks as we do for ResFCN (see Section II-B). From the segmentation results, we can see an improvement in quality from VGG-FCN to ResFCN. Moreover, on the Busy Parking Lot data set, the ResFCN also demonstrates a fairly strong ability to generalize to an "unseen" scene outside the training data set (see Fig. 9). However, there are some vehicles that cannot be separated in

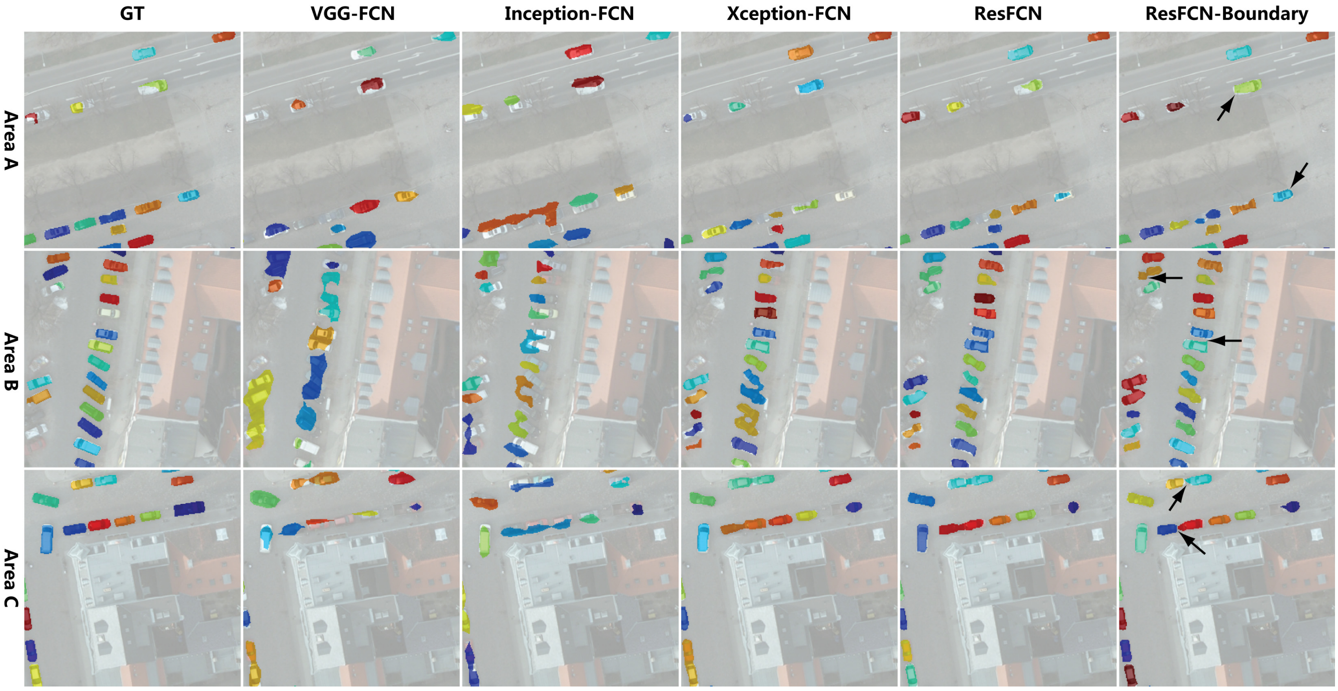


Fig. 7. Instance segmentation results of the ISPRS Potsdam data set. (From left to right) Ground truth, VGG-FCN, Inception-FCN, Xception-FCN, ResFCN, and B-ResFCN (different colors denote the individual vehicle objects). The three areas are derived from Fig. 4.

TABLE III
PIXEL-LEVEL OAs AND F1-SCORES FOR THE CAR CLASS ON THE ISPRS POTSDAM DATA SET

Model	OA	OA (eroded)	F1 score	F1 score (eroded)
ResFCN	99.79	99.89	93.43	95.66
B-ResFCN	99.79	99.89	93.44	95.87

both segmentation results produced using the aforementioned networks due to the extremely close vehicle distance. The situation is further deteriorated when the imagery suffers from the effects of shadow, as the cases shown in the zoomed-in areas of Fig. 9. On the other hand, to identify the role of the semantic boundary component of the proposed unified multitask learning network architecture, we also performed an ablation study to compare the performance of networks relying on the prediction of vehicles. In comparison with the ResFCN, the semantic B-ResFCN is able to separate those "touching" cars clearly, which qualitatively highlights the superiority of a semantic boundary-aware network by exploring the complementary information under a unified multitask learning network architecture. Fig. 8 shows a couple of example segmentations using the proposed B-ResFCN on several frames of the Busy Parking Lot data set.

D. Quantitative Evaluation

To verify the effectiveness of networks used, we reported the pixel-level OAs and F1 scores of the car class on our test set of the ISPRS Potsdam data set in Table III and compared with the state-of-the-art methods. These metrics are calculated on a full reference and an alternative ground truth obtained by eroding the boundaries of objects by a circular disk of 3 pixel radius. The current state-of-the-art CASIA2 (in the leaderboard http://www2.isprs.org/potsdam-2d-semantic-labeling.html) obtains the F1 score of 96.2% for the vehicle segmentation on the held-out test set (which is different from the validation set we use) using IRRG. Our B-ResFCN is competitive with the F1 score of 95.87% obtained by using the RGB information only on our own test set. This indicates that the trained network can be though as a good, competitive model for the follow-up experiments. Note that the pixelwise OA and F1 score can only evaluate the segmentation performance at a pixel level instead of instance level. Therefore, they are actually not suitable for our task.

To quantitatively evaluate the performance of different approaches for vehicle segmentation at the instance level, the evaluation criteria we use are instance-level F1 score, precision, recall, and Dice similarity coefficient. The first three criteria consider the performance of vehicle detection, and the last validates the performance of the instance-level segmentation.

1) Detection: For the vehicle detection evaluation, the metric instance-level F1 score[5] is employed, which is the harmonic mean of instance-level precision P and recall R, defined as

$$F1 = \frac{2PR}{P+R}, \quad P = \frac{N_{\mathrm{tp}}}{N_{\mathrm{tp}} + N_{\mathrm{fp}}}, \quad R = \frac{N_{\mathrm{tp}}}{N_{\mathrm{tp}} + N_{\mathrm{fn}}} \tag{7}$$

where N_{tp}, N_{fp}, and N_{fn} are the number of true positives, false positives, and false negatives, respectively. Here, the ground truth for each segmented vehicle is the object in the manually

[5] Note that the instance-level F1 score is different from the pixelwise F1 score used by the ISPRS semantic labeling evaluation (http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html).

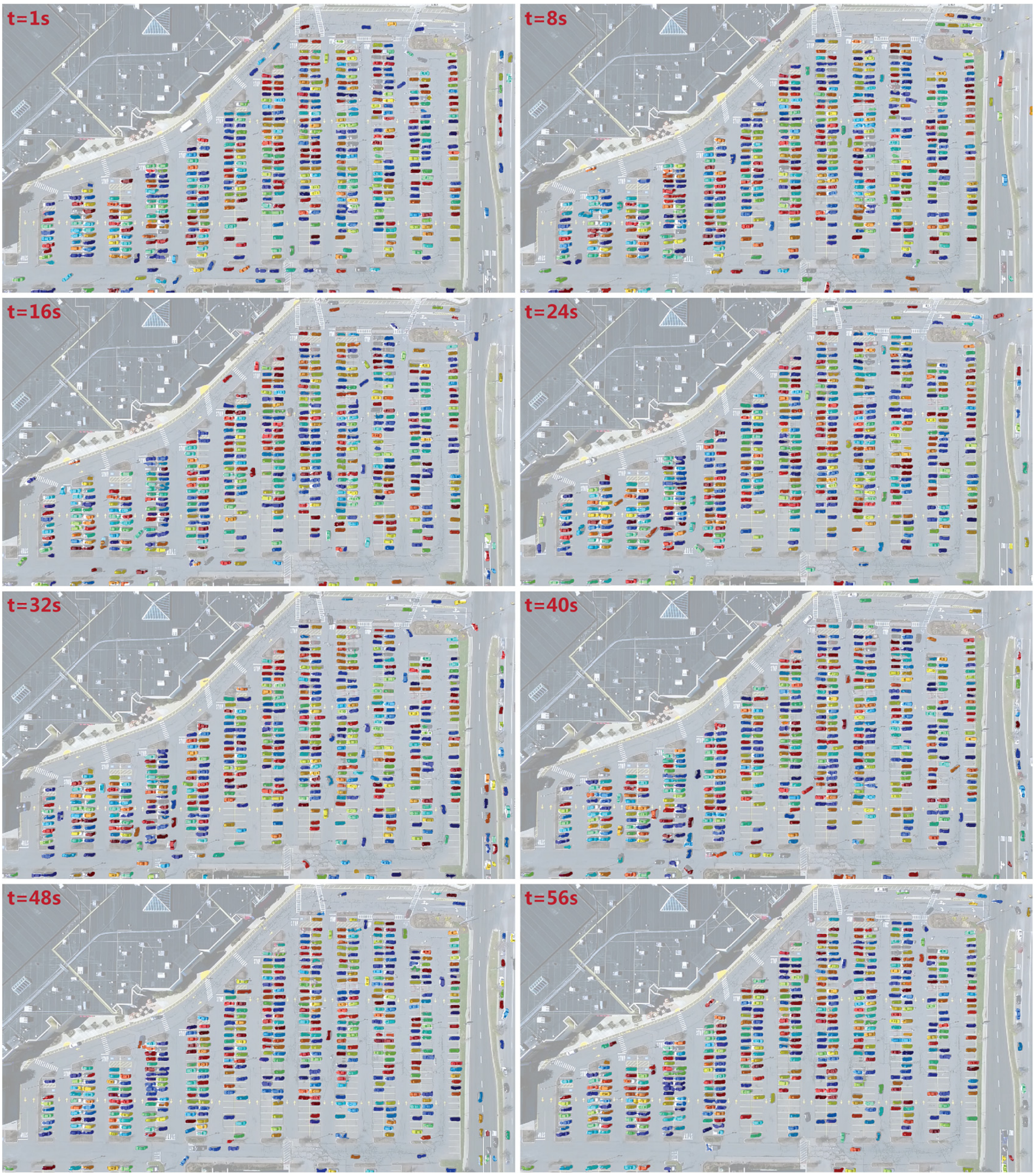


Fig. 8. Example segmentations using the proposed B-ResFCN in several frames of the Busy Parking Lot data set.

labeled segmentation mask that has a maximum overlap with the segmented vehicle. When calculating N_{tp} and N_{fp}, a segmented vehicle that intersects with at least 50% of its ground truth is considered as a true positive; otherwise, it is regarded as a false positive. For N_{fn}, a false negative indicates a ground-truth object that has less than 50% of its area overlapped by its corresponding segmented vehicle or has no corresponding segmented vehicle.

The detection results of different networks on the ISPRS Potsdam data set and the Busy Parking Lot scene are shown in Tables IV and V, respectively. Among the networks without a semantic boundary component, the ResFCN surpasses all other models (VGG-FCN, Inception-FCN, and Xception-FCN), highlighting the strength of residual learning-based FCN architecture with the multilevel contextual feature representations in our task. The network with the semantic boundary

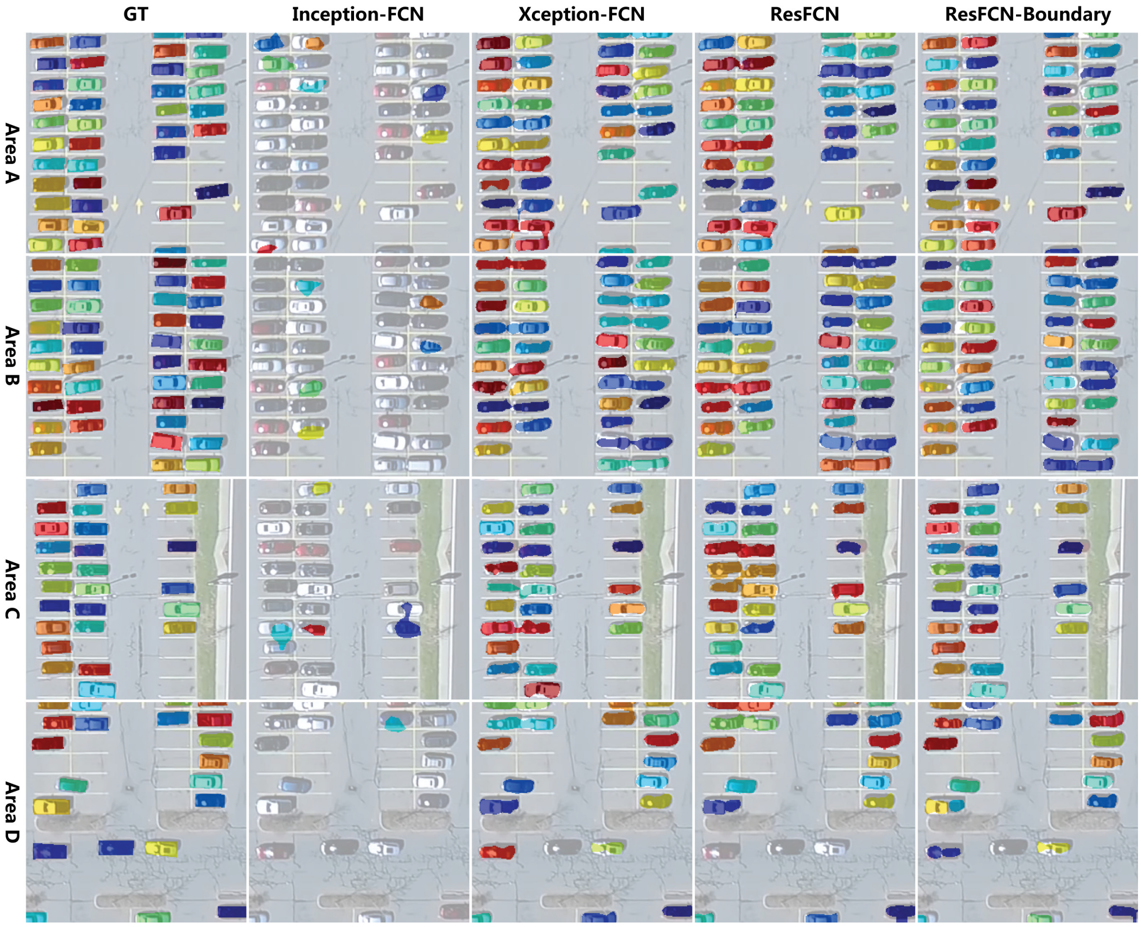


Fig. 9. Instance segmentation maps of the Busy Parking Lot data set. (From left to right) Ground truth, Inception-FCN, Xception-FCN, ResFCN, and B-ResFCN (different colors denote the individual vehicles). The four areas are derived from Fig. 6.

TABLE IV

Detection Results of Different Networks on the ISPRS Potsdam Semantic Labeling Data Set (Instance-Level F1 Score, Precision, and Recall)

Model	2_12			5_12			7_7			7_8			7_9		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R
VGG-FCN	66.04	70.00	62.50	57.00	61.45	53.14	59.21	61.95	56.70	57.21	66.84	50.00	61.31	65.91	57.31
B-VGG-FCN	70.27	68.42	72.22	69.85	67.42	72.47	71.03	68.47	73.79	67.96	66.86	69.09	66.47	60.96	73.08
Inception-FCN	51.91	55.45	48.80	31.65	37.42	27.42	40.00	43.41	37.08	27.79	31.70	24.74	40.87	45.02	37.42
B-Inception-FCN	55.15	50.61	60.58	46.14	47.42	44.92	53.81	52.91	54.75	43.47	42.45	44.54	50.74	47.49	54.47
Xception-FCN	96.92	98.21	95.65	83.55	81.11	86.14	93.33	94.59	92.11	92.05	93.10	91.01	93.92	96.59	91.40
B-Xception-FCN	97.00	**100**	94.17	88.40	**88.60**	88.19	93.65	96.47	91.00	93.58	97.54	89.94	94.63	97.50	91.92
ResFCN	97.93	**100**	95.93	83.88	80.84	87.15	94.72	96.86	92.67	**95.62**	**97.93**	**93.42**	95.25	96.23	**94.30**
B-ResFCN	**98.31**	**100**	**96.67**	**88.57**	87.08	**90.11**	**96.43**	**97.12**	**95.74**	95.19	97.88	92.64	**95.76**	**97.83**	93.77

TABLE V

Detection Results of Different Methods on the Proposed Busy Parking Lot UAV Video Data Set (Instance-Level F1 Score, Precision, and Recall)

Model	Frame@1s			Frame@15s			Frame@30s			Frame@45s			Frame@59s		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R
Inception-FCN	15.48	60.00	8.89	15.67	51.09	9.25	13.92	43.43	8.29	11.56	41.98	6.71	7.75	39.29	4.30
B-Inception-FCN	17.74	62.50	10.34	19.84	58.72	11.94	18.71	51.69	11.42	17.84	55.34	10.63	10.63	51.67	5.93
Xception-FCN	87.25	86.82	87.69	87.27	85.28	89.36	86.58	84.14	89.16	87.10	84.82	89.50	75.65	74.12	77.25
B-Xception-FCN	91.43	89.72	**93.20**	90.15	86.80	**93.78**	90.12	87.69	92.70	90.35	87.64	**93.22**	88.30	84.24	92.77
ResFCN	88.73	89.71	87.77	89.43	89.76	89.10	90.43	91.38	89.50	88.81	88.69	88.92	87.10	90.23	84.17
B-ResFCN	**93.29**	**95.16**	91.50	**92.55**	**91.52**	93.61	**93.62**	**94.02**	**93.22**	**93.06**	**94.33**	91.83	**94.54**	**95.28**	**93.81**

component—i.e., B-ResFCN—achieved the best results on most test images of the ISPRS Potsdam scene and surpassed the others by a significant margin on the Busy Parking Lot data set, demonstrating the effectiveness of the semantic boundary-aware multitask learning network in this instance segmentation problem. From Tables IV and V, we observe that all the

TABLE VI

SEGMENTATION RESULTS OF DIFFERENT METHODS ON THE BUSY PARKING LOT UAV VIDEO DATA SET (INSTANCE-LEVEL DICE SIMILARITY COEFFICIENT)

Model	**Frame@1s**	**Frame@15s**	**Frame@30s**	**Frame@45s**	**Frame@59s**
Inception-FCN	26.81	26.06	25.68	22.89	23.77
B-Inception-FCN	32.37	33.07	33.34	30.44	31.26
Xception-FCN	72.74	72.74	72.85	72.47	71.31
B-Xception-FCN	77.31	**77.50**	77.22	77.13	76.32
ResFCN	71.17	71.47	71.76	68.82	72.73
B-ResFCN	**78.84**	77.33	**79.13**	**77.83**	**79.39**

TABLE VII

SEGMENTATION RESULTS OF DIFFERENT METHODS ON THE ISPRS POTSDAM SEMANTIC LABELING DATA SET (INSTANCE-LEVEL DICE SIMILARITY COEFFICIENT)

Model	**2_12**	**5_12**	**7_7**	**7_8**	**7_9**
VGG-FCN	58.88	45.79	53.13	51.09	54.25
B-VGG-FCN	71.48	64.48	74.54	70.43	69.47
Inception-FCN	52.79	34.37	37.15	35.08	44.22
B-Inception-FCN	55.26	35.69	46.76	37.33	47.14
Xception-FCN	90.05	73.05	84.84	84.58	86.54
B-Xception-FCN	91.44	75.47	85.12	88.64	87.95
ResFCN	91.97	77.68	89.10	89.78	89.65
B-ResFCN	**93.80**	**77.72**	**90.61**	**91.19**	**90.66**

networks yield a fairly lower instance-level F1, precision, and recall on the Busy Parking Lot data set than on the ISPRS Potsdam data set. This mainly comes from the different difficulty levels of the two data sets. Specifically, high-density parking, strong light conditions, critical effects of shadow, and a slightly blurry image quality lead to the fact that networks have achieved a more inferior performance on the proposed data set than on the Potsdam scene.

2) Segmentation: The Dice similarity coefficient is often used to evaluate a segmentation performance. Given a set of pixels $\boldsymbol{V}$ denoted as a segmented vehicle and a set of pixels $\boldsymbol{G}$ annotated as a ground-truth object, the Dice similarity coefficient is defined as

$$D(\boldsymbol{V}, \boldsymbol{G}) = 2(|\boldsymbol{V} \cap \boldsymbol{G}|)/(|\boldsymbol{V}| + |\boldsymbol{G}|). \tag{8}$$

However, this is not suitable for segmentation evaluation on individual objects (i.e., instance segmentation). Instead, in this paper, an instance-level Dice similarity coefficient is defined and employed as

$$D_{\text{ins}}(\boldsymbol{V}, \boldsymbol{G}) = \frac{1}{2}\left[\sum_{i=1}^{N_V} \omega_i D(\boldsymbol{V}_i, \boldsymbol{G}_i) + \sum_{j=1}^{N_G} \tilde{\omega}_j D(\tilde{\boldsymbol{V}}_j, \tilde{\boldsymbol{G}}_j)\right] \tag{9}$$

where $\boldsymbol{V}_i$, $\boldsymbol{G}_i$, $\tilde{\boldsymbol{G}}_j$, and $\tilde{\boldsymbol{V}}_j$ are the ith segmented vehicle, the ground-truth object that maximally overlaps $\boldsymbol{V}_i$, the jth ground-truth object, and the segmented vehicle that maximally overlaps $\tilde{\boldsymbol{G}}_j$, respectively. N_V and N_G denote the total number of segmented vehicles and ground-truth objects, respectively. Furthermore, ω_i and $\tilde{\omega}_j$ are coefficients and can be calculated as

$$\omega_i = \frac{|\boldsymbol{V}_i|}{\sum_{k=1}^{N_V} |\boldsymbol{V}_k|}, \quad \tilde{\omega}_j = \frac{|\tilde{\boldsymbol{G}}_j|}{\sum_{k=1}^{N_G} |\tilde{\boldsymbol{G}}_k|}. \tag{10}$$

Tables VI and VII show the segmentation results of different approaches on the Potsdam scene and Busy Parking Lot data set, respectively. We can see that our B-ResFCN achieves the best performance on these two data sets. Compared with the ResFCN, there is a 1.16% increment in terms of the instance-level Dice similarity coefficient on the Potsdam data set and a 7.31% improvement on the Busy Parking Lot scene. From the figures in Tables VI and VII, we can see that the networks offer a more inferior performance on the Busy Parking Lot data set than on the Potsdam scene. This is also in line with our intention of proposing a more challenging benchmark data set for the vehicle instance segmentation problem. In addition, it is worth noting that basically all the networks with boundary components can offer better instance segmentations compared with those without boundary. This means that multitask learning is useful for different CNN variants in our task.

IV. CONCLUSION

In this paper, we propose a semantic boundary-aware unified multitask learning ResFCN in order to handle a novel problem (i.e., vehicle instance segmentation). In particular, the proposed network harnesses the multilevel contextual features learned from different residual blocks in a residual network architecture to produce better pixelwise likelihood maps. We theoretically analyze the reason behind this. Furthermore, our network creates two separate, yet identical branches to simultaneously predict the semantic segmentation masks of vehicles and semantic boundaries. The joint learning of these two problems is beneficial for separating "touching" vehicles which are often not correctly differentiated into instances. The network is validated using a large high-resolution aerial image data set, ISPRS Potsdam Semantic Labeling data set, and the proposed Busy Parking Lot UAV Video data set. To quantitatively evaluate the performance of different approaches for the vehicle instance segmentation, we advocate using an instance-level F1 score, precision, recall, and Dice similarity coefficient as evaluation criteria, instead of traditional pixelwise OA and F1 score for semantic segmentation. Both visual and quantitative analyses of the experimental results demonstrate the effectiveness of our approach.

ACKNOWLEDGMENT

The authors would like to thank the ISPRS for making the Potsdam data set available.

References

[1] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2016.

[2] L. Mou *et al.*, "Multitemporal very high resolution from space: Outcome of the 2016 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3435–3447, Aug. 2017.

[3] N. Audebert, B. Le Saux, and S. Lefèvre, "Fusion of heterogeneous data in convolutional networks for urban semantic labeling," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, Mar. 2017, pp. 1–4.

[4] L. Mou and X. X. Zhu. (2018). "RiFCN: Recurrent network in fully convolutional network for semantic segmentation of high resolution remote sensing images." [Online]. Available: https://arxiv.org/abs/1805.02091

[5] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Graph-based registration, change detection, and classification in very high resolution multitemporal remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 2940–2951, Jul. 2016.

[6] D. Wen, X. Huang, L. Zhang, and J. A. Benediktsson, "A novel automatic change detection method for urban high-resolution remotely sensed imagery based on multiindex scene representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 609–625, Jan. 2016.

[7] C. Wu, B. Du, X. Cui, and L. Zhang, "A post-classification change detection method based on iterative slow feature analysis and Bayesian soft fusion," *Remote Sens. Environ.*, vol. 199, pp. 241–255, Sep. 2017.

[8] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, p. 506, 2016.

[9] L. Mou and X. X. Zhu, "Spatiotemporal scene interpretation of space videos via deep neural network and tracklet analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1823–1826.

[10] G. Kopsiaftis and K. Karantzalos, "Vehicle detection and traffic density monitoring from very high resolution satellite video data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1881–1884.

[11] W. Shao, W. Yang, G. Liu, and J. Liu, "Car detection from high-resolution aerial imagery using multiple features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2012, pp. 4379–4382.

[12] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1635–1647, Mar. 2014.

[13] T. Moranduzzo and F. Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6356–6367, Oct. 2014.

[14] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[15] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[16] L. Mou and X. X. Zhu. (2018). "IM2HEIGHT: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network." [Online]. Available: https://arxiv.org/abs/1802.10249

[17] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.

[18] L. Mou, L. Bruzzone, and X. X. Zhu. (2018). "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery." [Online]. Available: https://arxiv.org/abs/1803.02642

[19] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[20] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[21] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, no. 4, p. 312, 2017.

[22] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sens.*, vol. 9, no. 4, p. 368, 2017.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla. (2015). "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." [Online]. Available: https://arxiv.org/abs/1511.00561

[24] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshop*, Jun. 2016.

[25] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2015, pp. 2650–2658.

[26] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1–10.

[27] Z. Wu, C. Shen, and A. van den Hengel. (2016). "High-performance semantic segmentation using very deep fully convolutional networks." [Online]. Available: https://arxiv.org/abs/1604.04339

[28] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1–9.

[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. (2016). "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs." [Online]. Available: https://arxiv.org/abs/1606.00915

[32] S. Zhang *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1–9.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, Sep. 2015, pp. 1–14.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 630–645.

[35] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1–9.

[36] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 519–534.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[38] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 818–833.

[39] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, p. 1.

[40] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1–10.

[41] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, Jan. 2018.

[42] F. Rottensteiner *et al.*, "The ISPRS benchmark on urban object classification and 3D building reconstruction," *ISPRS Ann. Photogramm., Remote Sensing Spatial Inf. Sci.*, vol. 1, no. 3, pp. 293–298, 2012.

[43] T. Dozat, *Incorporating Nesterov Momentum into Adam*. Accessed: Jun. 26, 2018. [Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf

[44] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1–9.

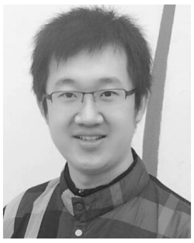
[45] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[46] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[47] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1–6.

[49] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1–8.



Lichao Mou (S’16) received the bachelor’s degree in automation from the Xi’an University of Posts and Telecommunications, Xi’an, China, in 2012, and the master’s degree in signal and information processing from the University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the German Aerospace Center, Wessling, Germany, and the Technical University of Munich, Munich, Germany.

In 2015, he was with the Computer Vision Group, University of Freiburg, Breisgau, Germany, for six months. His research interests include remote sensing, computer vision, and machine/deep learning, especially remote sensing video analysis and deep networks with their applications in remote sensing.

Mr. Mou was a recipient of the First Place at the 2016 IEEE GRSS Data Fusion Contest and a finalist for the Best Student Paper Award at the 2017 Joint Urban Remote Sensing Event.

Xiao Xiang Zhu (S’10–M’12–SM’14) received the M.Sc., Dr.Ing., and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She was a Guest Scientist or a Visiting Professor with the Italian National Research Council (CNR-IREA), Naples, Italy, in 2009, Fudan University, Shanghai, China, in 2014, The University of Tokyo, Tokyo, Japan, in 2015, and the University of California at Los Angeles, Los Angeles, CA, USA, in 2016. She is currently the Professor for Signal Processing in Earth Observation (SiPEO), TUM, and the German Aerospace Center (DLR), Wessling, Germany, and the Head of the Department of EO Data Science, Earth Observation Center, DLR, and the Helmholtz Young Investigator Group (SiPEO), DLR, and TUM. Her research interests include remote sensing and earth observation, signal processing, machine learning, and data science, with a special application focus on global urban mapping.

Dr. Zhu is a member of the Young Academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities, the German National Academy of Sciences Leopoldina, and the Bavarian Academy of Sciences and Humanities. She is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

F Mou L., Hua Y., Zhu X.X., 2019. A Relation-Augmented Fully Convolutional Network for Semantic Segmentation in Aerial Scenes, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

A Relation-Augmented Fully Convolutional Network for Semantic Segmentation in Aerial Scenes

Lichao Mou[1,2*], Yuansheng Hua[1,2*], Xiao Xiang Zhu[1,2]
[1] Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Germany
[2] Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Germany
{lichao.mou, yuansheng.hua, xiaoxiang.zhu}@dlr.de

Abstract

Most current semantic segmentation approaches fall back on deep convolutional neural networks (CNNs). However, their use of convolution operations with local receptive fields causes failures in modeling contextual spatial relations. Prior works have sought to address this issue by using graphical models or spatial propagation modules in networks. But such models often fail to capture long-range spatial relationships between entities, which leads to spatially fragmented predictions. Moreover, recent works have demonstrated that channel-wise information also acts a pivotal part in CNNs. In this work, we introduce two simple yet effective network units, the spatial relation module and the channel relation module, to learn and reason about global relationships between any two spatial positions or feature maps, and then produce relation-augmented feature representations. The spatial and channel relation modules are general and extensible, and can be used in a plug-and-play fashion with the existing fully convolutional network (FCN) framework. We evaluate relation module-equipped networks on semantic segmentation tasks using two aerial image datasets, which fundamentally depend on long-range spatial relational reasoning. The networks achieve very competitive results, bringing signicant improvements over baselines.

1. Introduction

Semantic segmentation of an image involves a problem of inferring every pixel in the image with the semantic category of the object to which it belongs. The emergence of deep convolutional neural networks (CNNs) [19, 33, 12, 16, 1, 40] and massive amounts of labeled data has brought significant progress in this direction. However, although with more complicated and deeper networks and more labeled samples, there is a technical hurdle in the application of CNNs to semantic image segmentation—contextual information.

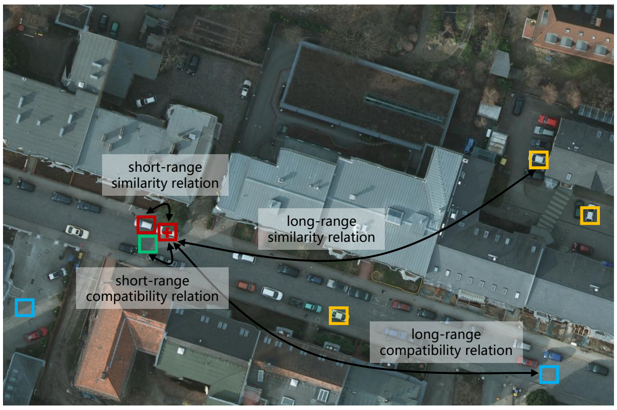


Figure 1: Illustration of long-range spatial relations in an aerial image. Appearance similarity or semantic compatibility between patches within a local region (red–red and red–green) and patches in remote regions (red–yellow and red–blue) underlines our global relation modeling.

It has been well recognized in the computer vision community for years that contextual information, or *relation*, is capable of offering important cues for semantic segmentation tasks [11, 39]. For instance, spatial relations can be considered semantic similarity relationships among regions in an image. In addition, spatial relations also involve compatibility and incompatibility relationships, *i.e.*, a vehicle is likely to be driven or parked on pavements, and a piece of lawn is unlikely to appear on the roof of a building. Unfortunately, only convolution layers cannot model such spatial relations due to their local valid receptive field[1].

Nevertheless, under some circumstances, spatial rela-

[*] Equal contribution

[1] Feature maps from deep CNNs like ResNet usually have large receptive fields due to deep architectures, whereas the study of [43] has shown that CNNs are apt to extract information mainly from smaller regions in receptive fields, which are called valid receptive fields.

tions are of paramount importance, particularly when a region in an image exhibits significant visual ambiguities. To address this issue, several attempts have been made to introduce spatial relations into networks by using either graphical models or spatial propagation networks. However, these methods seek to capture global spatial relations implicitly with a chain propagation way, whose effectiveness depends heavily on the learning effect of long-term memorization. Consequently, these models may not work well in some cases like aerial scenes (see Figure 5 and Figure 6), in which long-range spatial relations often exist (*cf.* Figure 1). Hence, explicit modeling of long-range relations may provide additional crucial information but still remains under-explored for semantic segmentation.

This work is inspired by the recent success of relation networks in visual question answering [31], object detection [13], and activity recognition in videos [42]. Being able to reason about relationships between entities is momentous for intelligent decision-making. A relation network is capable of inferring relationships between an individual entity (*e.g.*, a patch in an image) and a set of other entities (*e.g.*, all patches in the image) by agglomerating information. The relations vary at both long-range and short-range scales and are learned automatically, driven by tasks. Moreover, a relation network can model dependencies between entities, without making excessive assumptions on their feature distributions and locations.

In this work, our goal is to increase the representation capacity of a fully convolutional network (FCN) for semantic segmentation in aerial scenes by using relation modules: describing relationships between observations in convolved images and producing relation-augmented feature representations. Given that convolutions operate by blending spatial and cross-channel information together, we capture relations in both spatial and channel domains. More specifically, two plug-and-play modules—a spatial relation module and a channel relation module—are appended on top of feature maps of an FCN to learn different aspects of relations and then generate spatial relation-augmented and channel relation-augmented features, respectively, for semantic segmentation. By doing so, relationships between any two spatial positions or feature maps can be modeled and used to further enhance feature representations. Furthermore, we study empirically two ways of integrating two relation modules—serial and parallel.

Contributions. This work's contributions are threefold.

- We propose a simple yet effective and interpretable relation-augmented network that enables spatial and channel relational reasoning in networks for semantic segmentation on aerial imagery.
- A spatial relation module and a channel relation module are devised to explicitly model global relations, which are subsequently harnessed to produce spatial- and channel-augmented features.
- We validate the effectiveness of our relation modules through extensive ablation studies.

2. Related Work

Semantic segmentation of aerial imagery. Earlier studies [35] have focused on extracting useful low-level, hand-crafted visual features and/or modeling mid-level semantic features on local portions of images ([17, 26, 38, 27, 28, 44, 15] employ deep CNNs and have made a great leap towards end-to-end aerial image parsing. In addition, there are numerous contests aiming at semantic segmentation from overhead imagery recently, *e.g.*, Kaggle[2], SpaceNet[3], and DeepGlobal[4].

Graphical models. There are many graphical model-based methods being employed to achieve better semantic segmentation results. For example, the work in [5] makes use of a CRF as post-processing to improve the performance of semantic segmentation. [41] and [22] further make the CRF module differentiable and integrate it as a joint-trained part within networks. Moreover, low-level visual cues, *e.g.*, object contours, have also been considered structure information [3, 4]. These approaches, however, are sensitive to changes in appearance and expensive due to iterative inference processes required.

Spatial propagation networks. Learning spatial propagation with networks for semantic segmentation have attracted high interests in recent years. In [25], the authors try to predict entities of an affinity matrix directly by learning a CNN, which presents a good segmentation performance, while the affinity is followed by a nondifferentiable solver for spectral embedding, which results in the fact that the whole model cannot be trained end-to-end. The authors of [20] train a CNN model to learn a task-dependent affinity matrix by converting the modeling of affinity to learning a local linear spatial propagation. Several recent works [18, 21, 6] focus on the extension of this work. In [2, 29], spatial relations are modeled and reinforced via interlayer propagation. [2] proposes an Inside-Outside Net (ION) where four independent recurrent networks that move in four directions are used to pass information along rows or columns. [29] utilizes four slice-by-slice convolutions within feature maps, enabling message passings between neighboring rows and columns in a layer. The spatial propagation of these methods is serial in nature, and thus each position could only receive information from its neighbors.

[2] https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection

[3] https://spacenetchallenge.github.io/

[4] http://deepglobe.org/challenge.html

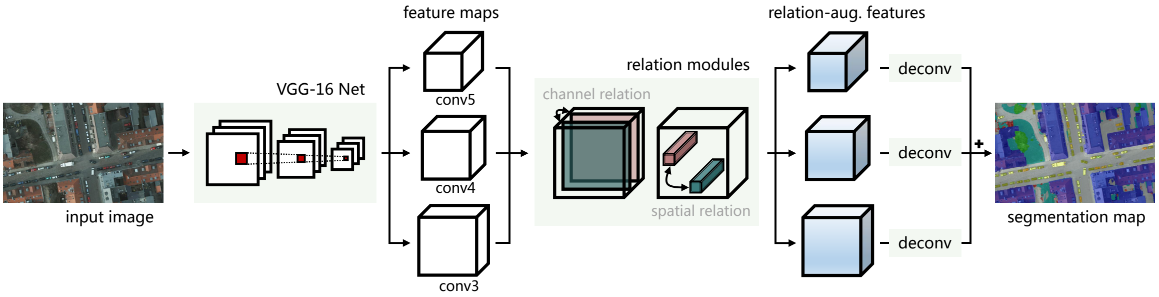


Figure 2: An overview of the relation module-equipped fully convolutional network.

Relation networks. Recently, the authors of [31] propose a relational reasoning network for the problem of visual question answering, and this network achieves a super-human performance. Later, [42] proposes a temporal relation network to enable multi-scale temporal relational reasoning in networks for video classification tasks. In [13], the authors propose an object relation module, which allows modeling relationships among sets of objects, for object detection tasks. Our work is motivated by the recent success of these works, but we focus on modeling spatial and channel relations in a CNN for semantic segmentation.

Unlike graphical model-based [9, 37] and spatial propagation network-based methods, we explicitly take spatial relations and channel relations into account, so that semantic image segmentation could benet from short- and long-range relational reasoning.

3. Our Approach

In this section, an overview of the proposed relational context-aware network is given to present a comprehensive picture. Afterwards, two key components, the spatial relation module and the channel relation module, are introduced, respectively. Finally, we describe the strategy of integrating these modules for semantic segmentation.

3.1. Overview

As illustrated in Fig. 2, the proposed network takes VGG-16 [34] as a backbone to extract multi-level features. Outputs of *conv3*, *conv4*, and *conv5* are fed into the channel and spatial relation modules (see Figure 2) for generating relation-augmented features. These features are subsequently fed into respective convolutional layers with 1×1 filters to squash the number of channels to the number of categories. Finally, the convolved feature maps are upsampled to a desired full resolution and element-wise added to generate final segmentation maps.

3.2. Spatial Relation Module

In order to capture global spatial relations, we employ a spatial relation module, where the spatial relation is defined as a composite function with the following equation:

$$\mathrm{SR}(\boldsymbol{x}_i, \boldsymbol{x}_j) = f_{\phi_s}(g_{\theta_s}(\boldsymbol{x}_i, \boldsymbol{x}_j)) \,. \tag{1}$$

Denote by $\boldsymbol{X} \in \mathbb{R}^{C\times H\times W}$ a random variable representing a set of feature maps. $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two feature-map vectors and identified by spatial positions indices i and j. The size of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is $C \times 1 \times 1$. To model a compact relationship between these two feature-map vectors, we make use of an embedding dot production as g_{θ_s} instead of a multilayer perceptron (MLP), and the latter is commonly used in relational reasoning modules [31, 42]. Particularly, g_{θ_s} is defined as follows:

$$g_{\theta_s}(\boldsymbol{x}_i, \boldsymbol{x}_j) = u_s(\boldsymbol{x}_i)^T v_s(\boldsymbol{x}_j) \,, \tag{2}$$

where $u_s(\boldsymbol{x}_i) = \boldsymbol{W}_{u_s}\boldsymbol{x}_i$ and $v_s(\boldsymbol{x}_j) = \boldsymbol{W}_{v_s}\boldsymbol{x}_j$. $\boldsymbol{W}_{u_s}$ and $\boldsymbol{W}_{v_s}$ are weight matrices and can be learned during the training phase. Considering computational efficiency, we realize Eq. (2) in matrix format with the following steps:

1. Feature maps $\boldsymbol{X}$ are fed into two convolutional layers with 1×1 filters to generate $u_s(\boldsymbol{X})$ and $v_s(\boldsymbol{X})$, respectively.
2. Then $u_s(\boldsymbol{X})$ and $v_s(\boldsymbol{X})$ are reshaped (and transposed) into $HW \times C$ and $C \times HW$, correspondingly.
3. Eventually, the matrix multiplication of $u_s(\boldsymbol{X})$ and $v_s(\boldsymbol{X})$ is conducted to produce a $HW \times HW$ matrix, which is further reshaped to form a spatial relation feature of size $HW \times H \times W$.

It is worth nothing that the spatial relation feature is not further synthesized (*e.g.*, summed up), as fine-grained contextual characteristics are essential in semantic segmentation tasks. Afterwards, we select the ReLU function as f_{ϕ_s} to eliminate negative spatial relations.

However, relying barely on spatial relations leads to a partial judgment. Therefore, we further blend the spatial relation feature and original feature maps $\boldsymbol{X}$ as follows:

$$\boldsymbol{X}_s = [\boldsymbol{X}, \mathrm{SR}(\boldsymbol{X})] \,. \tag{3}$$

Here we simply use a concatenation operation, i.e., $[\cdot, \cdot]$, to enhance original features with spatial relations. By doing so, output features are abundant in global spatial relations, while high-level semantic features are also preserved.

3.3. Channel Relation Module

Although the spatial relation module is capable of capturing global contextual dependencies for identifying various objects, misdiagnoses happen when objects share similar distribution patterns but vary in channel dimensionality. In addition, a recent work [14] has shown the benefit of enhancing channel encoding in a CNN for image classification tasks. Therefore, we propose a channel relation module to model channel relations, which can be used to enhance feature discriminabilities in the channel domain. Similar to the spatial relation module, we define the channel relation as a composite function with the following equation:

$$\mathrm{CR}(\boldsymbol{X}_p, \boldsymbol{X}_q) = f_{\phi_c}(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q)), \tag{4}$$

where the input is a set of feature maps $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_C\}$, and $\boldsymbol{X}_p$ as well as $\boldsymbol{X}_q$ represents the p-th and the q-th channels of $\boldsymbol{X}$. Embedding dot production is employed to be g_{θ_c}, defined as

$$g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q) = u_c(\mathrm{GAP}(\boldsymbol{X}_p))^T v_c(\mathrm{GAP}(\boldsymbol{X}_q)), \tag{5}$$

for capturing global relationships between feature map pairs, where $\mathrm{GAP}(\cdot)$ denotes the global average pooling function. Notably, considering that the preservation of spatial structural information distracts the analysis of channel inter-dependencies, we adopt averages of $\boldsymbol{X}_p$ and $\boldsymbol{X}_q$ as channel descriptors before performing dot production. More specifically, we feed feature maps into a global average pooling layer for generating a set of channel descriptors of size $C \times 1 \times 1$, and then exploit two convolutional layers with 1×1 filters to produce $u_c(\boldsymbol{X})$ and $v_c(\boldsymbol{X})$, respectively. Afterwards, an outer production is performed to generate a $C \times C$ channel relation feature, where the element located at (p, q) indicates $g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q)$.

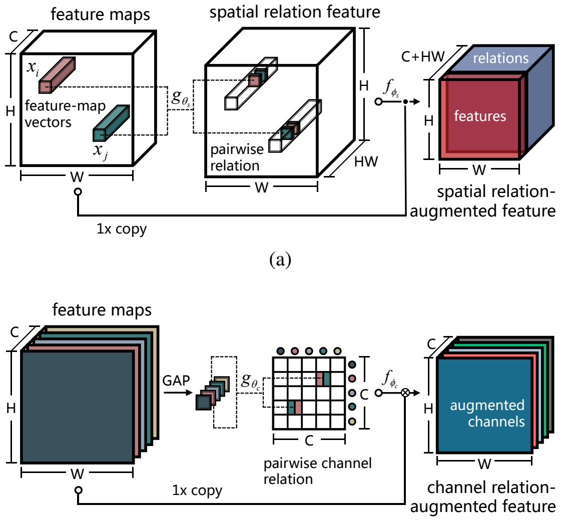
Furthermore, we emphasize class-relevant channel relations as well as suppress irrelevant channel dependencies by adopting a softmax function as f_{ϕ_c}, formulated as

$$f_{\phi_c}(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q)) = \frac{\exp(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q))}{\sum_{q=1}^{C} \exp(g_{\theta_c}(\boldsymbol{X}_p, \boldsymbol{X}_q))}, \tag{6}$$

where we take $\boldsymbol{X}_p$ as an example. Consequently, a discriminative channel relation map $\mathrm{CR}(\boldsymbol{X})$ can be obtained, where each element represents the corresponding pairwise channel relation.

To integrate $\mathrm{CR}(\boldsymbol{X})$ and original feature maps $\boldsymbol{X}$, we reshape $\boldsymbol{X}$ into a matrix of $C \times HW$ and employ a matrix multiplication as follows:

$$\boldsymbol{X}_c = \boldsymbol{X}^T \mathrm{CR}(\boldsymbol{X}). \tag{7}$$



(a)

(b)

Fig 3: Diagrams of (a) spatial relation module and (b) channel relation module.

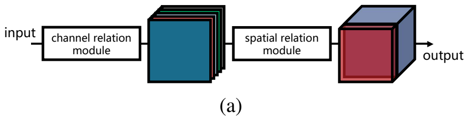
With this design, the input features are enhanced with channel relations and embedded with not only initial discriminative channel properties but also global inter-channel correlations. Eventually, $\boldsymbol{X}_c$ is reshaped to $C \times H \times W$ and fed into subsequent procedures.

3.4. Integration of Relation Modules

In order to jointly enjoy benefits from spatial and channel relation modules, we further aggregate features $\boldsymbol{X}_s$ and $\boldsymbol{X}_c$ to generate spatial and channel relation-augmented features. As shown in Fig. 4, we investigate two integration patterns, namely serial integration and parallel integration, to blend $\boldsymbol{X}_s$ and $\boldsymbol{X}_c$. For the former, we append the spatial relation module to the channel relation module and infer $\boldsymbol{X}_s$ from $\boldsymbol{X}_c$ instead of $\boldsymbol{X}$, as presented in Eq. (1) and Eq. (7). For the latter, spatial relation-augmented features and channel relation-augmented features are obtained simultaneously and then aggregated by performing concatenation. Influences of different strategies are discussed in Section 4.2.

4. Experiments

To verify the effectiveness of long-range relation modeling in our network, aerial image datasets are used in experiments. This is because aerial images are taken from nadir view, and the spatial distribution/relation of objects in these images is diverse and complicated, as shown in Figure 1. Thus, we perform experiments on two aerial image semantic segmentation datasets, *i.e.*, ISPRS Vaihingen and Potsdam datasets, and results are discussed in subsequent sections.



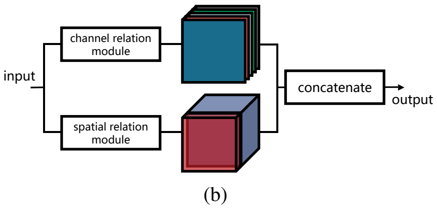


Fig 4: Two integration manners: (a) serial and (b) parallel.

4.1. Experimental Setup

Datasets. The Vaihingen dataset[5] is composed of 33 aerial images collected over a 1.38 km^2 area of the city, Vaihingen, with a spatial resolution of 9 cm. The average size of each image is 2494×2064 pixels, and each of them has three bands, corresponding to near infrared (NIR), red (R), and green (G) wavelengths. Notably, DSMs, which indicate the height of all object surfaces in an image, are also provided as complementary data. Among these images, 16 of them are manually annotated with pixel-wise labels, and each pixel is classified into one of six land cover classes. Following the setup in [24, 36, 32, 27], we select 11 images for training, and the remaining five images (image IDs: 11, 15, 28, 30, 34) are used to test our model.

The Potsdam dataset[6] consists of 38 high resolution aerial images, which covers an area of 3.42 km^2, and each aerial image is captured in four channels (NIR, R, G, and blue (B)). The size of all images is 6000×6000 pixels, which are annotated with pixels-level labels of six classes as the Vaihingen dataset. The spatial resolution is 5 cm, and coregistered DSMs are available as well. To train and evaluate networks, we utilize 10 images for training and build the test set with the remaining images (image IDs: 02_11, 02_12, 04_10, 05_11, 06_07, 07_08, 07_10), which follows the setup in [24, 32].

Implementation. The proposed network is initialized with separate strategies with respect to two dominant components: the feature extraction module is initialized with CNNs pre-trained on ImageNet dataset [7], while convolutional layers in relation modules are initialized with a Glorot uniform initializer. Notably, weights in the feature extraction module are trainable and fine-tuned during the training phase.

Regarding the used optimizer, we choose Nestrov Adam [8] and set parameters of the optimizer as recommended: $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 1\text{e}{-08}$. The learning rate is initialized as $2\text{e}{-04}$ and decayed by 0.1 when validation loss is saturated. The loss of our network is simply defined as categorical cross-entropy. We implement the network on TensorFlow and train it on one NVIDIA Tesla P100 16GB GPU for 250k iterations. The size of the training batch is 5, and we stop training when the validation loss fails to decrease.

Table 1: Ablation Study on the Vaihingen Dataset.

Model Name	crm	srm	mean F_1	OA
Baseline FCN [23]			83.74	86.51
RA-FCN-crm	✓		87.24	88.38
RA-FCN-srm		✓	88.36	89.03
P-RA-FCN	✓	✓	88.50	89.18
S-RA-FCN	✓	✓	**88.54**	**89.23**

[1] RA-FCN indicates the proposed relation-augmented FCN.
[2] crm indicates the channel relation module.
[3] srm indicates the spatial relation module.
[4] P-RA-FCN indicates that crm and srm are appended on top of the backbone in parallel.
[5] S-RA-FCN indicates that crm is followed by srm.

Evaluation metric. To evaluate the performance of networks, we calculate F_1 score with the following formula:

$$F_1 = (1+\beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}, \quad \beta = 1, \quad (8)$$

for each category. Furthermore, mean F_1 score is computed by averaging all F_1 scores to assess models impartially. Notably, a large F_1 score suggests a better result. Besides, mean IoU (mIoU) and overall accuracy (OA) that indicates overall pixel accuracy, are also calculated for a comprehensive comparison with different models.

4.2. An Ablation Study for Relation Modules

In our network, spatial and channel relation modules are employed to explore global relations in both spatial and channel domains. To validate the effectiveness of these modules, we perform ablation experiments (*cf.* Table 1). Particularly, instead of being utilized simultaneously, spatial and channel relation modules are embedded on top of the backbone (i.e., VGG-16), respectively. Besides, we also discuss different integration strategies (*i.e.*, parallel and serial) of relation modules in Table 1.

The ablation experiments are conducted on the Vaihingen dataset. As can be seen in Table 1, relation modules bring a significant improvement as compared to the baseline FCN (VGG-16), and various integration schemes lead

[5] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html
[6] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html

Table 2: Experimental Results on the Vaihingen Dataset

Model Name	Imp. surf.	Build.	Low veg.	Tree	Car	mean F_1	mIoU	OA
SVL-boosting+CRF* [10]	86.10	90.90	77.60	84.90	59.90	79.90	-	84.70
RF+dCRF* [30]	86.90	92.00	78.3	86.90	29.00	74.60	-	85.90
CNN-FPL* [36]	-	-	-	-	-	83.58	-	87.83
FCN [23]	88.67	92.83	76.32	86.67	74.21	83.74	72.69	86.51
FCN-dCRF [5]	88.80	92.99	76.58	86.78	71.75	83.38	72.28	86.65
SCNN [29]	88.21	91.80	77.17	87.23	78.60	84.40	73.73	86.43
Dilated FCN [5]	90.19	94.49	77.69	87.24	76.77	85.28	-	87.70
FCN-FR* [24]	**91.69**	**95.24**	79.44	88.12	78.42	86.58	-	88.92
PSPNet (VGG16) [40]	89.92	94.36	78.19	87.12	72.97	84.51	73.97	87.62
RotEqNet* [27]	89.50	94.80	77.50	86.50	72.60	84.18	-	87.50
RA-FCN-srm	91.01	94.86	80.01	88.74	87.16	88.36	79.48	89.03
P-RA-FCN	91.46	95.02	80.40	88.56	**87.08**	88.50	79.72	89.18
S-RA-FCN	91.47	94.97	**80.63**	**88.57**	87.05	**88.54**	**79.76**	**89.23**

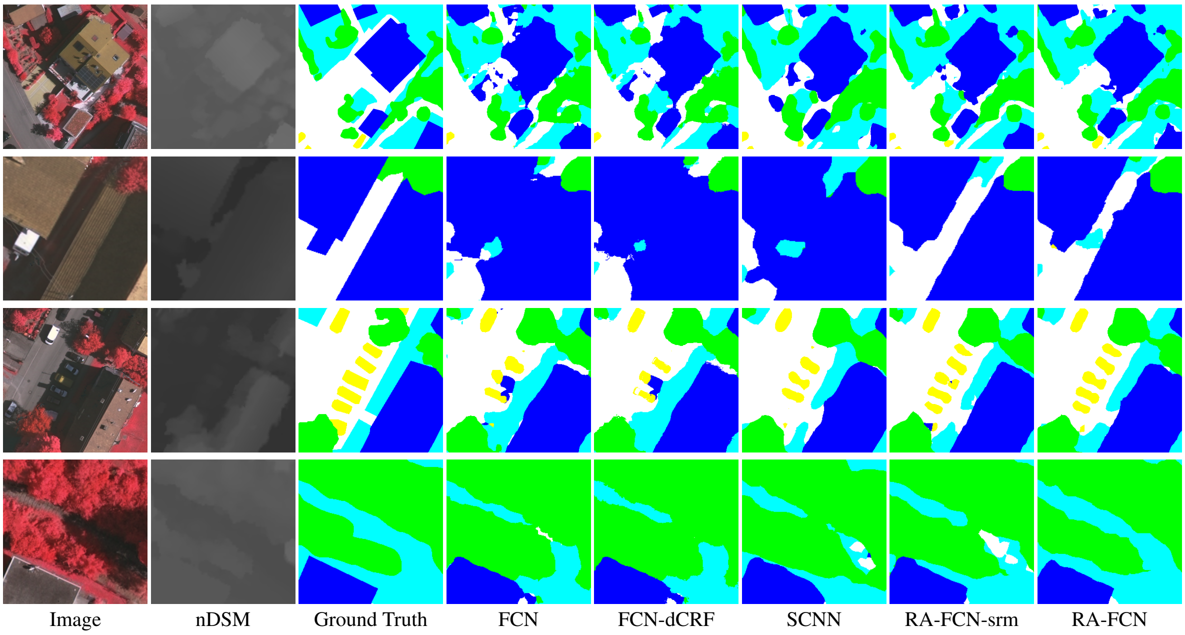


Figure 5: Examples of segmentation results on the Vaihingen dataset. Legend—white: impervious surfaces, blue: buildings, cyan: low vegetation, green: trees, yellow: cars.

to a slight influence on the performance of our network. In detailed, the use of only the channel relation module yields a result of 87.24% in the mean F_1 score, which brings a 3.50% improvement. Meanwhile, RA-FCN with only the spatial relation module outperforms the baseline by a 4.62% gain in the mean F_1 score. In addition, we note that squeeze-and-excitation module [14] can also model dependencies between channels. However, in our experiments, the proposed channel relation module performs better.

Moreover, by taking advantage of spatial relation-augmented and channel relation-augmented features simultaneously, the performance of our network is further boosted up. The parallel integration of relation modules brings increments of 1.26% and 0.14% in the mean F_1 score with respect to RA-FCN-crm and RA-FCN-srm. Besides, a serial aggregation strategy is discussed, and results demonstrate that it behaves superiorly as compared to other models. To be more specific, such design achieves the highest mean F_1 score, 88.54%, as well as the highest overall accuracy, 89.23%. To conclude, spatial- and channel-augmented

Table 3: Numerical Results on the Potsdam Dataset

Model Name	Imp. surf.	Build.	Low veg.	Tree	Car	Clutter	mean F_1	mIoU	OA
FCN [23]	88.61	93.29	83.29	79.83	93.02	69.77	84.63	78.34	85.59
FCN-dCRF [5]	88.62	93.29	83.29	79.83	93.03	69.79	84.64	78.35	85.60
SCNN [29]	88.37	92.32	83.68	80.94	91.17	68.86	84.22	77.72	85.57
Dilated FCN* [5]	86.52	90.78	83.01	78.41	90.42	68.67	82.94	-	84.14
FCN-FR* [24]	89.31	94.37	84.83	81.10	93.56	76.54	86.62	-	87.02
RA-FCN-srm	90.48	93.74	85.67	83.10	94.34	74.02	86.89	81.23	87.61
P-RA-FCN	90.92	94.20	86.64	83.00	94.44	**77.88**	87.85	81.85	88.30
S-RA-FCN	**91.33**	**94.70**	**86.81**	**83.47**	**94.52**	77.27	**88.01**	**82.38**	**88.59**

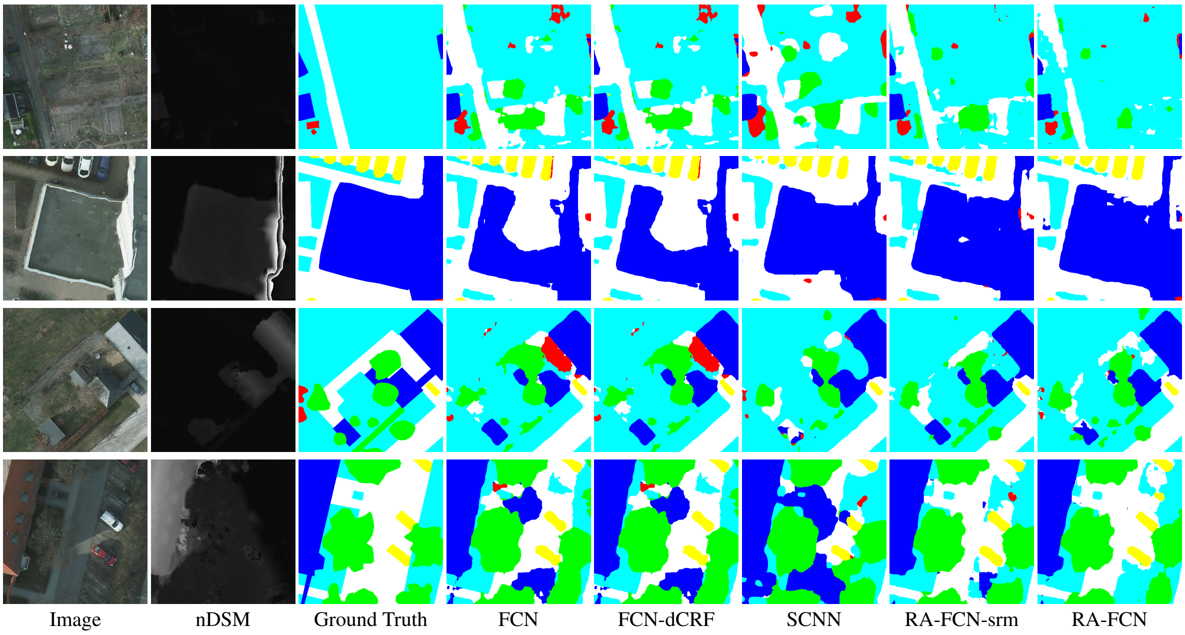


Figure 6: Examples of segmentation results on the Potsdam dataset. Legend—white: impervious surfaces, blue: buildings, cyan: low vegetation, green: trees, yellow: cars, red: clutter/background.

features extracted from relation modules carry out not only high-level semantics but also global relations in spatial and channel dimensionalities, which reinforces the performance of a network for semantic segmentation in aerial scenes.

4.3. Comparing with Existing Works

For a comprehensive evaluation, we compare our model with six existing methods, including FCN [23], FCN with fully connected CRF (FCN-dCRF) [5], spatial propagation CNN (SCNN) [29], FCN with atrous convolution (Dilated FCN) [5], FCN with feature rearrangement (FCN-FR) [24], CNN with full patch labeling by learned upsampling (CNN-FPL) [36], RotEqNet [27], PSPNet with VGG16 as backbone [40], and several traditional methods [10, 30].

Numerical results on the Vaihingen dataset are shown in Table 2. It is demonstrated that RA-FCN outperforms other methods in terms of mean F_1 score, mean IoU, and overall accuracy. Specifically, comparisons with FCN-dCRF and SCNN, where RA-FCN-srm obtains increments of 4.98% and 3.69% in mean F_1 score, respectively, validate the high performance of the spatial relation module in our network. Besides, compared to FCN-FR, RA-FCN reaches improvements of 1.96% and 1.57% in mean F_1 score and overall accuracy, which indicates the effectiveness of integrating the spatial relation module and channel relation module. Furthermore, per-class F_1 scores are calculated to assess the performance of recognizing different objects. It is noteworthy that our method remarkably surpasses other competitors in identifying scattered cars for its capacity of capturing long-range spatial relation.

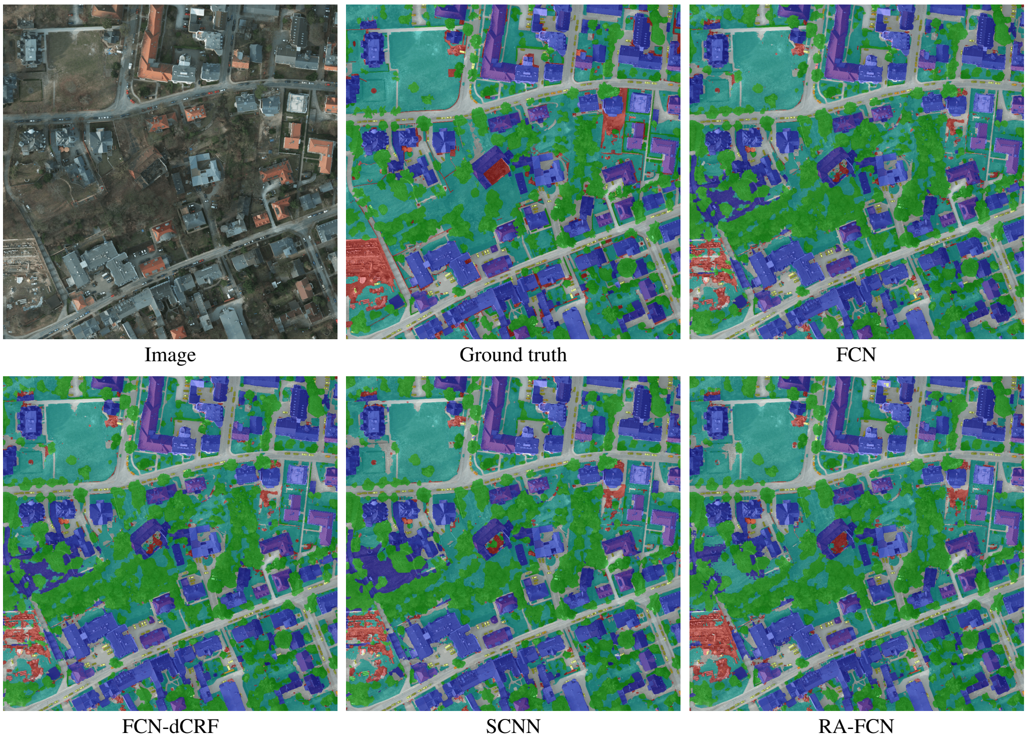


Fig 7: Example segmentation results of an image in the test set on Potsdam dataset ($90,000\ \mathrm{m}^2$). Legend—white: impervious surfaces, blue: buildings, cyan: low vegetation, green: trees, yellow: cars, red: clutter/background. Zoom in for details.

4.4. Qualitative Results

Fig. 5 shows a few examples of segmentation results. The second row demonstrates that networks with local receptive fields or relying on fully connected CRFs and spatial propagation modules fail to recognize impervious surfaces between two buildings, whereas our models make relatively accurate predictions. This is mainly because in this scene, the appearance of impervious surfaces is highly similar to that of the right building, which leads to a misjudgment of rival models. Thanks to the spatial relation module, RA-FCN-srm or RA-FCN is able to effectively capture useful visual cues from more remote regions in the image for an accurate inference. Besides, examples in the third row illustrate that RA-FCN is capable of identifying dispersively distributed objects as expected.

4.5. Results on the Potsdam Dataset

In order to further validate the effectiveness of our network, we conduct experiments on the Potsdam dataset, and numerical results are shown in Table 3. The spatial relation module contributes to improvements of 2.25% and 2.67% in the mean F_1 score with respect to FCN-dCRF and SCNN, and the serial integration of both relation modules brings increments of 1.39% and 1.54% in the mean F_1 score, mean IoU, and overall accuracy, respectively.

Moreover, qualitative results are presented in Figure 6. As shown in the first row, although low vegetation regions comprise intricate local contextual information and are liable to be misidentified, RA-FCN obtains more accurate results in comparison with other methods due to its remarkable capacity of exploiting global relations to solve visual ambiguities. The fourth row illustrates that outliers, i.e., the misclassified part of the building, can be eliminated by RA-FCN, while it is not easy for other competitors. To provide a thorough view of the performance of our network, we also exhibit a large-scale aerial scene as well as semantic segmentation results in Figure 7.

5. Conclusion

In this paper, we have introduced two effective network modules, namely the spatial relation module and the channel relation module, to enable relational reasoning in networks for semantic segmentation in aerial scenes. The comprehensive ablation experiments on aerial datasets where long-range spatial relations exist suggest that both relation modules have learned global relation information between objects and feature maps. However, our understanding of how these relation modules work for segmentation problems is preliminary and left as future works.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] G. Bertasius, J. Shi, and L. Torresani. Semantic segmentation with boundary neural fields. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv:1606.00915*, 2016.

[6] X. Cheng, P. Wang, and R. Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *European Conference on Computer Vision (ECCV)*, 2018.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[8] T. Dozat. Incorporating Nesterov momentum into Adam. 2015.

[9] N. Friedman and D. Koller. Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1-2):95–125, 2003.

[10] M. Gerke. *Use of the Stair Vision Library within the ISPRS 2D Semantic Labeling Benchmark (Vaihingen)*. 2015.

[11] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] Y. Hua, L. Mou, and X. X. Zhu. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:188–199, 2019.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[18] T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu. Adaptive affinity fields for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[20] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[21] S. Liu, G. Zhong, S. De Mello, J. Gu, V. Jampani, M.-H. Yang, and J. Kautz. Switchable temporal propagation network. In *European Conference on Computer Vision (ECCV)*, 2018.

[22] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. High-resolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7092–7103, 2017.

[25] M. Maire, T. Narihira, and S. X. Yu. Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] D. Marcos, D. Tuia, B. Kellenberger, L. Zhang, M. Bai, R. Liao, and R. Urtasun. Learning deep structured active contours end-to-end. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] D. Marcos, M. Volpi, B. Kellenberger, and D. Tuia. Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:96–107, 2018.

[28] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135:158–172, 2018.

[29] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial as deep: Spatial CNN for traffic scene understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[30] N. Quang, N. Thuy, D. Sang, and H. Binh. An efficient framework for pixel-wise building segmentation from aerial

images. In *International Symposium on Information and Communication Technology, ACM*, 2015.

[31] A. Santoro, D. Raposo, D. G.T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[32] J. Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv:1606.02585*, 2016.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *IEEE International Conference on Learning Representation (ICLR)*, 2015.

[35] P. Tokarczyk, J. D. Wegner, S. Walk, and K. Schindler. Features, color spaces, and boosting: New insights on semantic classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):280–295, 2015.

[36] M. Volpi and D. Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, 2017.

[37] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[38] S. Wang, M. Bai, G. Mattyus, H. Chen, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. TorontoCity: Seeing the world with a million eyes. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[39] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[42] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision (ECCV)*, 2018.

[43] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *IEEE International Conference on Learning Representation (ICLR)*, 2015.

[44] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.

G Related Publications

G.1 Journals

- Mou L., Zhu X. 2019. Learning to Pay Attention on Spectral Domain: A Spectral Attention Module-Based Convolutional Network for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, in press.
- Qiu C., Mou L., Schmitt M., Zhu X. 2019. Local Climate Zone-based Urban Land Cover Classification from Multi-seasonal Sentinel-2 Images with a Recurrent Residual Network, ISPRS Journal of Photogrammetry and Remote Sensing, in press.
- Li Q., Mou L., Xu Q., Zhang Y., Zhu X., 2019. R^3-Net: A Deep Network for Multi-oriented Vehicle Detection in Aerial Images and Videos, IEEE Transactions on Geoscience and Remote Sensing, in press.
- Mou L., Bruzzone L., Zhu X., 2019. Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery, IEEE Transactions on Geoscience and Remote Sensing, 57(2), 924-935.
- Hua Y., Mou L., Zhu X., 2019. Recurrently Exploring Class-wise Attention in A Hybrid Convolutional and Bidirectional LSTM Network for Multi-label Aerial Image Classification, ISPRS Journal of Photogrammetry and Remote Sensing, 149, 188-199.
- Qiu C., Schmitt M., Mou L., Ghamisi P., Zhu X., 2018. Feature Importance Analysis for Local Climate Zone Classification using a Residual Convolutional Neural Network with Multi-source Datasets, Remote Sensing, 10(10), 1572.
- Lyu H., Lu H., Mou L., Li W., Wright J., Li Xuecao, Li, Xinlu, Zhu X., Wang J., Yu L., Gong P., 2018. Long-Term Annual Mapping of Four Cities on Different Continents by Applying a Deep Information Learning Method to Landsat Data, Remote Sensing, 10(3), 471.
- Li Q., Mou L., Liu Q., Zhu X., 2018. HSF-Net: Multi-Scale Deep Feature Embedding for Ship Detection in Optical Remote Sensing Imagery, IEEE Transactions on Geoscience and Remote Sensing, 56(12), 7147-7161.

- Mou L., Zhu X., 2018. Vehicle Instance Segmentation from Aerial Image and Video Using a Multi-Task Learning Residual Fully Convolutional Network, IEEE Transactions on Geoscience and Remote Sensing, 56(11), 6699-6711.
- Hughes L., Schmitt M., Mou L., Wang Y., Zhu X., 2018. Identifying Corresponding Patches in SAR and Optical Images with a Pseudo-Siamese CNN, IEEE Geoscience and Remote Sensing Letters, 15(5), 784-788.
- Mou L., Ghamisi P., Zhu X., 2018. Unsupervised Spectral-Spatial Feature Learning via Deep Residual Conv-Deconv Network for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, 56(1), 391-406.
- Mou L., Ghamisi P., Zhu X., 2017. Deep Recurrent Neural Networks for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, 55(7), 3639-3655.
- Zhu X., Tuia D., Mou L., Xia G., Zhang L., Xu F., Fraundorfer F., 2017. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources, IEEE Geoscience and Remote Sensing Magazine, 5(4), 8-36.
- Mou L., Zhu X., Vakalopoulou M., Karantzalos K., Paragios N., Le Saux B., Moser G., Tuia D., 2017. Multi-temporal very high resolution from space: Outcome of the 2016 IEEE GRSS Data Fusion Contest, IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 10(8), 3435-3447.

G.2 Conferences

- Mou L., Hua Y., Zhu X.X., 2019. A Relation-augmented Fully Convolutional Network for Semantic Segmentation in Aerial Scenes, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
- Mou L., Hua Y., Zhu X.X., 2019. Spatial Relational Reasoning in Networks for Improving Semantic Segmentation of Aerial Images, in: IEEE International Geoscience and Remote Sensing Symposium (IGARSS).
- Hua Y., Mou L., Zhu X.X., 2019. Label Relation Inference for mMlti-label Aerial Image Classification, in: IEEE International Geoscience and Remote Sensing Symposium (IGARSS).
- Liu L., Mou L., Zhu X.X., Mandal M., 2019. Skin lesion segmentation based on improved U-Net, in: IEEE Canadian Conference on Electrical Computer Engineering (CCECE).
- Hua Y., Mou L., Zhu X.X., 2019. Multi-label Aerial Image Classification using A Bidirectional Class-wise Attention Network, in: Joint Urban Remote Sensing Event (JURSE).

- Hua Y., Mou L., Zhu X.X., 2018. LahNet: A Convolutional Neural Network Fusing Low- and High-Level Features for Aerial Scene Classification, in: IEEE International Geoscience and Remote Sensing Symposium (IGARSS).
- Sun Y., Hua Y., Mou L., Zhu X.X., 2019. Large-scale Building Height Estimation from Single VHR SAR image Using Fully Convolutional Network and GIS building footprints, in: Joint Urban Remote Sensing Event (JURSE).
- Hu, J., Mou, L., Schmitt, A., Zhu, X. 2017. FusioNet: A two-stream convolutional neural network for urban scene classification using PolSAR and hyperspectral data. In Urban Remote Sensing Event (JURSE), 2017 Joint.
- Mou, L., Schmitt, M., Wang, Y., Zhu, X. 2017. A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes. In Urban Remote Sensing Event (JURSE), 2017 Joint.
- Huang R., Taubenböck H., Mou L., Zhu X.X., 2018. Classification of Settlement Types from Tweets Using LDA and LSTM, in: 2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018. pp. 6408–6411.
- Li Q., Mou L., Jiang K., Liu Q., Wang Y., Zhu X.X., 2018. Hierarchical Region Based Convolution Neural Network for Multiscale Object Detection in Remote Sensing Images, in: 2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018. pp. 4355–4358. https://doi.org/10.1109/IGARSS.2018.8518345
- Mou L., Zhu X.X., 2018. A Recurrent Convolutional Neural Network for Land Cover Change Detection in Multispectral Images, in: 2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018. pp. 4363–4366.
- Qiu C., Schmitt M., Mou L., Zhu X.X., 2018. Urban Local Climate Zone Classification with a Residual Convolutional Neural Network and Multi-Seasonal Sentinel-2 Images, in: Proc. 10th IAPRS Workshop on Pattern Recognition in Remote Sensing.
- Qiu C. P., Schmitt M., Ghamisi P., Mou L., Zhu X., 2018. Feature importance analysis of Sentinel-2 imagery for large-scale Urban Local Climate Zone classification. In Geoscience and Remote Sensing Symposium (IGARSS) 2018, Valencia, Spain.
- Wang, Y., Zhu, X., Montazeri, S., Kang, J., Mou, L., Schmitt, M. 2017. Potential of the "SARptical" system. In Proceedings of FRINGE2017 Workshop. Helsinki, Finland.
- Mou, L., Schmitt, M., Wang, Y., Zhu, X. 2017. Identifying corresponding patches in SAR and optical imagery with a convolutional neural network. In IEEE Inter-

national Geoscience and Remote Sensing Symposium (IGARSS) 2017, Fort Worth, USA.

- Mou, L., Ghamisi, P., Zhu, X. 2017. Fully conv-deconv network for unsupervised spectral-spatial feature extraction of hyperspectral imagery via residual learning. In IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2017, Fort Worth, USA.
- Mou, L., Zhu, X., 2016. Spatiotemporal Scene Interpretation of Space Videos via Deep Neural Network and Tracklet Analysis, in: IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2016, Beijing, China.