Visualization-Based Active Learning for the Annotation of SAR Images

Mohammadreza Babaee, *Student Member, IEEE*, Stefanos Tsoukalas, Gerhard Rigoll, *Senior Member, IEEE*, and Mihai Datcu, *Fellow Member, IEEE*

***Abstract*—Active learning has gained a high amount of attention due to its ability to label a vast amount of unlabeled collected earth observation (EO) data. In this paper, we propose a novel active learning algorithm which is mainly based on employing a low-rank classifier as the training model and introducing a visualization support data point selection, namely, first certain wrong labeled (FCWL). The training model is composed of the logistic regression loss function and the trace-norm of learning parameters as regularizer. FCWL selects those data points whose labels are predicted wrong but the classifier is highly certain about them. Our experimental results performed on different extracted features from a dataset of SAR images confirm at least 10% improvement over the state-of-the-art methods.**

***Index Terms*—Active learning, synthetic aperture radar (SAR), trace-norm regularized classifier, visualization.**

I. Introduction

TODAY, we are dealing with a phenomenon, the so-called big data, where the amount of collected data has been increasing exponentially since the last decade [1]. For instance, the volume of collected earth observation (EO) data is in the order of several terra-bytes per day. Additionally, the complexity of the data is very high such that in most cases, very high-dimensional features are used to represent the data. Automatic storage and retrieval of this data require large-scale learning algorithms, which need a large set of labeled data to train. On one hand, providing labeled data is expensive and time consuming. On the other hand, the unlabeled data are available in a large scale for free and cheap. Therefore, active learning has gained high attention due to its ability in labeling the data and training the classifier simultaneously [2]–[6].

The works in [2]–[4] give an overview of some popular active learning methods. Each active learning method uses a different heuristic to select the most informative sample for labeling. Some popular heuristics are empirical risk reduction, uncertainty sampling, query by committee, expected model change, and spatial structure-based sample selection. Persello *et al.* [5] argue that an active learning algorithm should also take the different annotation costs for each sample in account, as is the case in a dynamic system. They address this issue in the framework of a Markov decision process, which tries to minimize the cumulative cost of training. In [6], a method is proposed, which combines the popular uncertainty criterion of the SVM classifier with structure information, gained by training a self-organizing map on the dataset.

In an active learning system, first, a small set of labeled data is used to train a multiclass classifier (e.g., SVM). Then, a subset of unlabeled data along with their predicted label is selected to query the true labels from the user. The user examines the predicted labels and relabels the data points (if necessary) and adds them to the pool of labeled data for retraining the classifier. This loop continues until all unlabeled data are labeled and also the classifier is trained. One challenge in active learning, which has gained much attention from researchers, is to select the most informative samples for labeling. However, as the amount of data increases, the performance of the classifier improves only slowly. In other words, the challenge of sample selection is important only when the amount of training data is low. When labeled data grows, the main issue in active learning switches to overfitting of the optimization of the classifier.

In this paper, we propose a novel active learning framework to annotate the massive amount of synthetic aperture radar (SAR) image repositories. Precisely, we employ a recently proposed classifier for large-scale datasets, as training model, to tackle the problem of overfitting. Our proposed algorithm is based on the work in [7] and [8]. In [7], an algorithm for large-scale multiclass image classification is proposed, which adds a trace-norm regularization penalty to the $L2$ regularized multiclass logistic loss function in order to avoid overfitting of the learning parameters. Furthermore, the objective is transformed into a convex function, which can be minimized with a coordinate-descent algorithm. Additionally, we introduce a visualization-based sample selection algorithm to query the labels. This method, the so-called first certain wrong labeled (FCWL), is based on a ranked list of the sample images, ordered by confidence. The images are visualized in tabular form, where each column of the table represents one of the categories of the dataset. The algorithm places the images predicted in each category in the corresponding column and sorts them by decreasing classifier confidence. The classifier confidence is measured by the distance of the sample to the class separating hyperplane, which is defined by the corresponding classifier parameters W. After the images have been presented, the user looks for the first wrong labeled image in one class and relabels it. Here, the user in each interaction corrects one sample, whose label is predicted

Manuscript received October 01, 2014; revised November 07, 2014; accepted December 24, 2014. This work was supported by Munich Aerospace.

M. Babaee, S. Tsoukala, and G. Rigoll are with the Institute for Human-Machine Communication, Technische Universität München, 80333 Munich, Germany (e-mail: reza.babaee@tum.de; s.tsoukalas@mytum.de; rigoll@tum.de).

M. Datcu is with the Remote Sensing Technology Institute (IMF), German Aerospace Center, Oberpfaffenhofen D-82234, Germany (e-mail: mihai.datcu@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/JSTARS.2015.2388496

incorrectly with high confidence by the classifier. This has the effect of introducing the maximum correction to the training model in each iteration. Precisely, the higher the classifier confidence is about a sample, the further away this sample is from the hyperplane defined by the classifier parameters. This means that the classifier hyperplane has to move the most in order to predict a different label for this sample which is equivalent to introducing a high correction to the classifier parameters. The process continues until a desirable accuracy of the classifier is achieved.

The main contributions of our work are as follows:

1) employing a trace-norm regularizer classifier as the training model in an active learning framework to annotate the SAR images.
2) studying in depth the trace-norm classifier.
3) introducing a novel active learning algorithm based on visualization.
4) conducting experiments to study the performance of the proposed active learning algorithm.

The rest of the paper is organized as follows. In Section II, we review the state-of-the-art active learning algorithms. We illustrate the concept of multiclass classification with the trace-norm regularized classifier in Section III. Here, we first study the proposed classifier and compare it in depth with support vector machines (SVMs). Then, we formulate one active learning algorithm solely based on this classifier. Finally, in Section IV, we explain our visualization-based active learning algorithm. Section V demonstrates our experiments conducted on real SAR image datasets. Finally, we draw our conclusion and describe future works in Section VI.

II. Related Work

Active learning is a widely used approach for the annotation of multimedia and EO data [2], [3], [9]. Active learning frameworks could be categorized in three different groups, namely: 1) membership query synthesis; 2) stream-based selective sampling; and 3) pool-based active learning [2].

Membership query synthesis is the first scenario considered in active learning [2]. In this scenario, the learning program might query any instance from the input space, included synthesized samples, which do not actually exist in the training data. In stream-based selective sampling, the samples are often drawn from the dataset one at a time. Here, the main challenge is how to decide whether to query a sample or not. One popular method for doing this is based on the prediction certainty of a trained classifier for this sample, e.g., by defining a certainty threshold and querying all samples, whose certainty is below that threshold [2].

Pool-based active learning, as the most common active learning framework, is based on the assumption that a big pool of unlabeled data $\mathcal{U}$ is available for free and that all data can be accessed at the same time [2]. This is the case in many real-world applications (e.g., image or document classification and EO data annotation). The goal in pool-based active learning is to select the most informative samples from the unlabeled data and add it to the pool of labeled data $\mathcal{L}$. Therefore, different criteria for optimum experimental design (OED) have led to the development of different active learning algorithms. For instance, OED [10] selects the points based on their spatial distribution. A-optimal design defines a cost function by the trace of the covariance matrix of the classifier parameters W and iteratively selects samples that minimize the value of this cost function. Transductive experimental design (TED) [11] minimizes the average predictive variance of the classifier on a test set. Finally, manifold adaptive experimental design (MAED) [12] extends the TED algorithm with a manifold adaptive kernel.

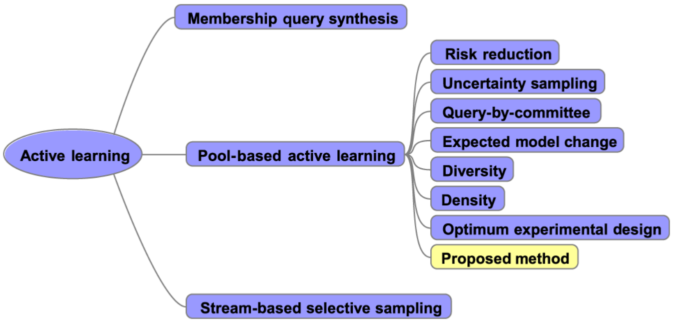


Fig. 1. Overview of active learning scenarios and sample selection strategies.

In addition to treating the active learning as an experimental design, there are some works which select the points based on other criteria such as diversity [13], risk reduction [14]–[16], reconstruction error [17], uncertainty [8], [18], [19], density [20], or a combination of several criteria [21]. For example, an active learning algorithm, the so-called $\mathrm{LLR}_{\mathrm{Active}}$, has been proposed based on locally linear reconstruction of data points, which shows impressive results when the dataset is coming from a low-dimensional manifold embedded in a high-dimensional space [17]. This method is inspired by a popular manifold learning algorithm, the so-called locally linear embedding [22], in which each point is reconstructed by a linear combination of its neighbor points. In this method, the most informative sample points are those points that can be better reconstructed by their neighbor points. Some works consider the selection of points from high density regions. Another popular active learning algorithm is $\mathrm{SVM}_{\mathrm{Active}}$[8], which selects samples for labeling based on their distance to the boundary of an SVM classifier [23]. This algorithm fits into the framework of uncertainty sampling [24], which involves selecting samples based on the uncertainty of a trained classifier. Fig. 1 provides an overview of the different active learning scenarios.

In addition to different sample selection strategies, different classification algorithms can be applied in an active learning framework. Many different classification algorithms have been used, such as k-NN, SVM, Gaussian mixture model [25], maximum entropy classifier [26], and graph-based semi-supervised learning [27].

In remote sensing, active learning has been intensively used to annotate the data [28]–[30]. The majority of works use SVM as classifier and use different aforementioned selection strategies to query the labels.

III. Trace-Norm Regularized Classifier (TC)

The motivation for using a low-rank regularizer in classification comes from the observation that when the SVM classifier is trained on large datasets, the singular values of the learning parameters (matrix W) have an exponential decay [7]. By looking at the dimensions of W, this can either be interpreted as the samples lying on subspace of lower dimension or as the classes being linear combinations of a smaller set of underlying prototype classes. Therefore, Harchaoui *et al.* [7] have proposed to leverage this property (i.e., the singular values of W have an exponential decay) by minimizing the rank of the matrix W and therefore keeping only the singular values with high magnitude.

We consider the set of n feature vectors $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of dimension d and the corresponding class labels $\mathcal{Y} = \{y_1, \ldots, y_n\}$ with a total number of k classes. The general linear multiclass classification problem involves learning a classifier $g(x) = \mathrm{argmax}_{l=1,\ldots,k} \boldsymbol{w}_l^{\mathrm{T}} x$ that predicts a class l, for each point $\boldsymbol{x}$. The classifier is specified by the k weight vectors $\boldsymbol{w}_l$. The general procedure to train the classifier involves minimizing the regularized empirical risk function

$$\min_{\boldsymbol{W} \in \mathbb{R}^{dxk}} \lambda \Omega(\boldsymbol{W}) + \frac{1}{n} \sum_{i=i}^{n} L(\boldsymbol{W}; \boldsymbol{x}_i, y_i) \tag{1}$$

with the regularization penalty Ω, the loss function L, and the weight matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$. One popular method for training the classifier is the one-versus-rest (OVR) strategy [31], which involves splitting the problem into l binary classification problems, where for each class, the labels corresponding to this class are set to 1 and the labels corresponding to other classes to -1. The binary problems can then be solved by an SVM classifier [23]. The first problem, which arises when trying to apply the low-rank regularizer to the OVR optimization problem, is that each of the columns of W is trained independently on the corresponding binary problem, while the low-rank constraint requires to treat the matrix $\boldsymbol{W}$ as a whole. Therefore, Harchaoui *et al.* [7] proposed to use the multinomial logistic loss function, which treats all classes simultaneously. Introducing the low-rank enforcing penalty and the multinomial logistic loss function into (1) leads to the following objective function for minimization:

$$\min_{\boldsymbol{W} \in \mathbb{R}^{dxk}} \lambda_1 \mathrm{rank}\,(\boldsymbol{W}) + \lambda_2 \|\boldsymbol{W}\|_2^2 + R_n(\boldsymbol{W}) \tag{2}$$

with

$$R_n(\boldsymbol{W}) = \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{W}; \boldsymbol{x}_i, y_i) \tag{3}$$

and

$$L(\boldsymbol{W}; \boldsymbol{x}, y) = \log\left(1 + \sum_{l \in \mathcal{Y} \setminus \{y\}} \exp\left\{\boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{x} - \boldsymbol{w}_y^{\mathrm{T}} \boldsymbol{x}\right\}\right). \tag{4}$$

This is a nonsmooth nonconvex optimization problem, which is difficult to solve. However, in [7], a solution is provided, where the low-rank penalty is replaced by its convex surrogate the trace-norm. The whole algorithm is summarized in Algorithm 1.

Algorithm 1. Solving trace-norm-regularized optimization problem (Summary of Algorithm in [7])

```
  Input:  regularization parameters $\lambda_1$ and $\lambda_2$ initial point
          $\boldsymbol{W}_{\boldsymbol{\theta}_0}$, convergence threshold $\epsilon$ training points $\mathcal{X}$
          and labels $\mathcal{Y}$
  Output: $\epsilon$-optimal $\boldsymbol{W}_{\boldsymbol{\theta}}$
Algorithm:

  for t = 0, 1, 2, ... do
    Compute top singular vector pair $\{\boldsymbol{u}_t, \boldsymbol{v}_t\}$ of $-\nabla \tilde{R}_n(\boldsymbol{W}_t)$
    Let $g_t = \lambda_1 + (\nabla \tilde{R}_n(W_t), \boldsymbol{u}_t \boldsymbol{v}_t^{\mathrm{T}})$
    if $g_t \leq -\epsilon/2$ then
      $W_{t+1} = W_t + \delta \boldsymbol{u}_t \boldsymbol{v}_t^{\mathrm{T}}$ with $\delta$ found by line-search
      $\theta_{t+1} = \delta$
      $\boldsymbol{\theta}_{t+1} = [\boldsymbol{\theta}_t, \theta_{t+1}]$
    else
      Check stopping conditions:
        $\forall i \in \mathcal{I} : \frac{\partial \tilde{R}_n(\boldsymbol{\theta})}{\partial \theta_i} + \lambda_1 \geq -\epsilon$
        $\forall i \in \mathcal{I} | \theta_i \neq 0 : \left| \frac{\partial \tilde{R}_n(\boldsymbol{\theta})}{\partial \theta_i} + \lambda_1 \right| \leq \epsilon$
      if stopping conditions satisfied then
        stop and return $\boldsymbol{W}_t$
      else
        Compute $\boldsymbol{\theta}_{t+1}$ as a solution of the following restricted
        problem:
        $\min_{\theta_1,\ldots,\theta_s} \lambda_1 \sum_{j=1}^{s} \theta_j + \tilde{R}_n\left(\sum_{j=1}^{s} \theta_j \boldsymbol{u}_j \boldsymbol{v}_j^{\mathrm{T}}\right)$
        subject to $\theta_j \geq 0,\ j = 1, \ldots, s$
      end if
    end if
  end for
```

The algorithm so far has only been described for the linear case. Even though it is possible to introduce a high-dimensional mapping for datasets with nonlinear mappings, as shown for SVM, this method is not used here. The reason is that the trace-norm regularized classifier was specifically developed for real datasets with high-dimensional feature spaces. In state-of-the-art methods high-dimensional image descriptors are used in combination with linear classifiers [7].

A. Comparison to the SVM Algorithm

The introduced low-rank base classifier has some similarities to the SVM algorithm. Both estimate the classification boundary through the matrix $\boldsymbol{W}$ by minimizing the regularized loss over the set of samples. However, there are also some important differences. These are related to the way multiclass problems are treated, to the type of loss functions and regularizers used to the existence of support vectors.

1) Multiclass Problem: The first difference is that the SVM algorithm cannot deal with multiple classes, simultaneously, and therefore has to resort to the OVR strategy, which involves solving multiple binary training problems, simultaneously. One problem of this approach is the resulting binary problems are imbalanced. By increasing the number of classes, the degree of imbalance increases. The second problem is that the matrix $\boldsymbol{W}$ is not treated as a whole. Instead, each column of $\boldsymbol{W}$ is estimated independently by training on the specific problem. Thus, the matrix $\boldsymbol{W}$ cannot be regularized as a whole and its columns might be imbalanced relative to each other.

2) Loss Function and Regularizer: The next difference is the type of loss function and regularizers used by the two algorithms. The SVM algorithm is a constrained optimization problem with the hinge loss function [32]. The hinge loss for the vector $\boldsymbol{w}$ and a sample $\boldsymbol{x}_i$ with label y_i is defined as

$$l_h(\boldsymbol{w}) = \max\{0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i\} \tag{5}$$

with the help of the hinge loss, the SVM optimization problem can be transformed into the following unconstrained optimization problem [32]:

$$\min_{\boldsymbol{w}} \lambda \|\boldsymbol{w}\|^2 + \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i\}. \tag{6}$$

This makes it possible to highlight the differences between the two classifiers more clearly. The TC uses a weighted combination of the L_2 norm and the trace-norm of $\boldsymbol{W}$. Additionally, as mentioned in SVM, the L_2 norm of each column of $\boldsymbol{W}$ is minimized independently, while for the TC, the two norms are minimized as a whole. Regarding the loss function, SVM uses the nonsmooth hinge loss function for the binary problem of whether a sample belongs to that class or not. Instead, the TC uses the multiclass logistic loss function, which is smooth and considers the difference of a sample class to all other classes, simultaneously.

3) Support Vectors: Another difference between the two classifiers is the existence of support vectors. In the SVM classifier, due to the nonsmooth hinge loss function only a subset of the vectors contributes to the training of the classifier. These vectors are the support vectors. The existence of support vectors makes it possible to reduce the training time of the SVM significantly, since the computations have to be performed only over this subset of vectors. In the TC, support vectors do not exist explicitly, due to the smooth multiclass logistic loss function. However, it might be desirable to find an approximation of support vectors for some applications, e.g., reduction of the number of samples for training. In the following, an approach for approximating the support vectors is presented.

The approach is based on the hinge loss function of the SVM classifier (5). Let y_i denote the true label of sample $\boldsymbol{x}_i$. Then, for each column $\boldsymbol{w}_l$ of $\boldsymbol{W}$, the hinge loss is computed by

$$l_{h;l} = \max\{0, 1 - \hat{y}_i \boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{x}_i\} \tag{7}$$

where $\hat{y}_i = 1$, if $l = y_i$ and $\hat{y}_i = -1$, otherwise. If any of the resulting hinge loss terms is greater than zero, i.e.,

$$l_{h;l} \geq 0, \quad l = 1, \ldots, k \tag{8}$$

then sample $\boldsymbol{x}_i$ is selected as support vectors of TC. Fig. 2 shows an example of the support vectors computed from a synthetic dataset. The dataset consists of three classes, with 100 samples per class. The positions of the samples are computed by Gaussian distributions. The support vectors are denoted by black crosses. Fig. 2(a) shows the support vectors of the SVM classifier and Fig. 2(b) shows the support vectors of the TC computed by inserting the columns of $\boldsymbol{W}$ into the hinge loss function. In the SVM classifier, the support vectors come up due to the nonsmooth loss function. The number of support vectors is controlled by the regularization parameter and with an increasing amount of support vectors over-fitting occurs. In the TC classifier, the smooth multiclass logistic loss function is used instead, which takes all samples into account and therefore by default has no support vectors. The usage of this loss function is required in order to use the trace-norm regularization term, which treats the matrix W as a whole. This is not possible with the OVR approach and the usage of the SVM hinge loss, which treats the multiclass problem as multiple binary problems. However, even though the TC classifier takes all samples into account, the over-fitting problem here is avoided with the added trace-norm regularization term. The disadvantage of taking all samples in account in the TC classifier is the increased computation time; however, a higher accuracy is achieved through this. The support vectors shown in Fig. 2 for the TC classifier are an approximation, which are estimated by introducing the hinge loss function into the matrix computed by the TC classifier and can be used in order to determine which vectors have the highest contribution to the optimization problem.

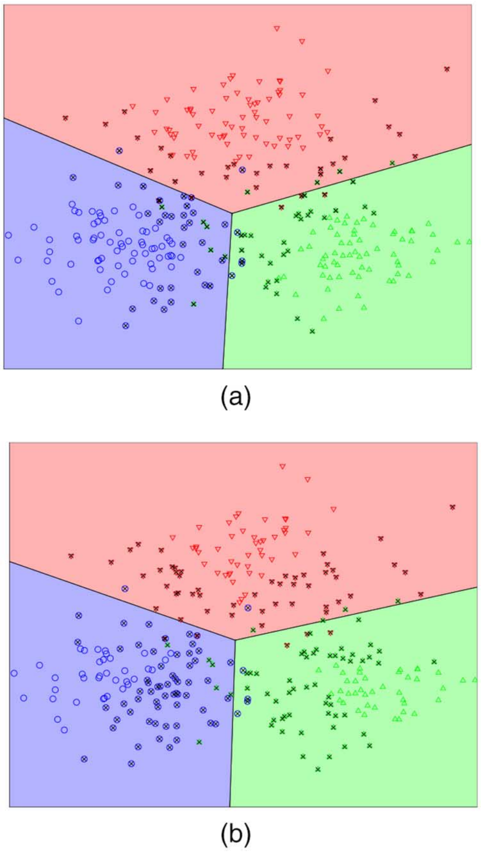


Fig. 2. Visualization of support vectors for a synthetic dataset. Black crosses denote the support vectors. (a) Support vectors of SVM classifier. (b) Support vectors of the TC, approximated by hinge loss function. In the TC, more features are considered as support vectors.

4) Computational Complexity: The computational complexity is an important factor for the comparison of the SVM

TABLE I
COMPUTATIONAL COMPLEXITY AND ACTUAL TRAINING TIMES OF CLASSIFIERS ON SAR DATASET REPRESENTED BY BoW OF SIFT FEATURE DESCRIPTORS

Classifier	SVM	TC (MATLAB)	TC (C++)
Time complexity	$O(kndp)$	$O(kndq)$	$O(kndq)$
Runtimes (s)	0.262	386.228	23.220

to the TC. For the solution of the SVM optimization problem, many different optimization techniques have been developed over the years. Therefore, it is difficult to estimate the overall computational complexity. For the LibSVM library [33], which is the SVM implementation used in this work, the overall computational complexity has been estimated to be $O(ndp)$ for the binary problem, where n is the number of samples, d is the dimension of the feature vectors, and p is the number of iterations. For the extension of the binary problem to the multiclass problem with k classes, the binary problem has to be solved k times in total. This leads to an overall time complexity of $O(kndp)$.

For the TC, the most expensive steps are the computation of $\nabla\tilde{R}_n(\boldsymbol{W})$ and of the top singular vector pair of $\nabla\tilde{R}_n(\boldsymbol{W})$. The top singular vector pair can be computed by the Lanczos method [34], which has a time complexity of $O(dk)$. The computation of $\nabla\tilde{R}_n(\boldsymbol{W})$ involves two steps: the matrix products $\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{x}_i$, which have a computational complexity of $O(ndk)$ and the summation of the logistic function over all terms, which has a computational complexity of $O(nk)$. Thus, the overall computational complexity of the TC is also in the order of $O(kndq)$, with q denoting the number of training iterations. It should be noted, however, that the TC requires more training iterations in total and more computations are performed during each training iteration. In this work, the training of the trace-norm regularized classifier was implemented on the scientific programming package MATLAB. The most expensive calculations, like $\nabla\tilde{R}_n(\boldsymbol{W})$, have been implemented in C++ in the format of mex files. Table I summarizes the training complexities and the measured training times on a real dataset for the different implementations. The real dataset consists of 500 samples of the SAR dataset, grouped in 15 classes. Each image is represented by the Bag-of-Word (BoW) model of SIFT local descriptors extracted from images.

B. Active Learning With TC

The TC can be employed in an active learning framework as the training model. For instance, it can be based on iteratively training the classifier on the available subset of labeled samples and selecting as the next sample for labeling, the point which is closest to the current boundary of the classifier, similar to the $\mathrm{SVM}_{\mathrm{Active}}$ algorithm [8]. Let $\mathcal{X}$ denote the set of all available samples and $\mathcal{L}$ denote the set of labeled samples. With the current weight matrix $W = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$, the margin of each sample is

$$\mu_i = \max_{l=1,\ldots,k} \boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{x}_i. \tag{9}$$

The index of the next sample for labeling is then given by

$$i_l = \underset{i}{\operatorname{argmin}}\{\mu_i | \boldsymbol{x}_i \in \mathcal{X}\backslash\mathcal{L}\}. \tag{10}$$

Training the classifier with each new labeled sample can be done efficiently by storing the matrix $\boldsymbol{W}$ and setting it as the starting point for the next training iteration. The whole active learning algorithm is summarized in Algorithm 2.

Algorithm 2. Active learning with TC

```
Input:  training points $\mathcal{X}$ and labels $\mathcal{Y}$ initial set of labeled
        samples $\mathcal{L}_0$ total number of points to label $m$
Output: new set of labeled samples $\mathcal{L}$ weight matrix $\boldsymbol{W}$
Algorithm:

for t = 0, 1, 2, . . . do
    obtain $\boldsymbol{W}_t$ by applying Algorithm 1 on $\mathcal{L}_t$ with initial
    weight matrix $\boldsymbol{W}_{t-1}$
    if $|\mathcal{L}_t| = m$ then
        stop and return $\boldsymbol{W}_t$, $\mathcal{L}_t$
    end if
    Compute margin $\mu_i$ for each sample according to (11)
    Select point $\boldsymbol{x}_i$ with smallest margin according to (10) and
    set $\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{\boldsymbol{x}_i\}$
end for
```

IV. VISUALIZATION-BASED INSTANCE SELECTION

In this section, a novel active learning method is introduced, which is based on the principles of uncertainty and expected model change. Compared to existing active learning algorithms, the difference here is that the sample selection strategy is also coupled to the visualization method in addition to the classifier. In the visualization, a ranked list of predictions ordered by confidence is shown to the user and the user is asked to select the first incorrectly predicted sample from one class. The algorithm can be used in combination with any classifier, for which a confidence metric can be computed, including the SVM and the TC introduced in the Section III.

For the description of the algorithm, we consider the set of n samples with feature vectors $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of dimension d and the corresponding class labels $\mathcal{Y} = \{y_1, \ldots, y_n\}$ with a total number of k classes. Additionally, let $\mathcal{L}$ denote the set of samples whose correct label is available to the algorithm.

The main idea of this algorithm is to select samples for labeling, which introduce the highest change into the model of a trained classifier. This is achieved by letting the algorithm predict labels during each iteration and asking the user to correct a label, which the algorithm is certain about, but predicts incorrectly. As the measure of certainty, we use the extension of the margin as suggested in the $\mathrm{SVM}_{\mathrm{Active}}$ algorithm [8] to a multiclass classifier. Given the current weight matrix $\boldsymbol{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$ of the classifier, we define the margin μ_i of each sample as

$$\mu_i = \max_{l=1,\ldots,k} \mathbf{w}_l^{\mathrm{T}}\mathbf{x}_i \tag{11}$$

TABLE II
PREDICTED SAMPLES IMAGES PRESENTED TO THE USER DURING EACH ITERATION

$\tilde{l}=1$	$\tilde{l}=2$	$\cdots$	$\tilde{l}=k$
$I_{\tilde{1};1}$	$I_{\tilde{2};1}$		$I_{\tilde{k};1}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$
$I_{\tilde{1};n_1}$	$I_{\tilde{2};n_2}$		$I_{\tilde{k};n_k}$

and the predicted label $\tilde{y}_i$ of each sample as

$$\tilde{y}_i = \arg\max_{l=1,\ldots,k} \mathbf{w}_l^{\mathrm{T}} \mathbf{x}_i. \tag{12}$$

Let $I_{\tilde{l};i}$ denote the image of the unlabeled sample $\mathbf{x}_{\tilde{l};i} \in \mathcal{X}\backslash\mathcal{L}$, predicted with label $\tilde{l}$. Then the algorithm during each iteration arranges these images in a table with increasing margins for each class, i.e.,

$$i \geq j \Leftrightarrow \mu_{\tilde{l};i} \geq \mu_{\tilde{l};j} \tag{13}$$

as suggested in Table II and lets the user select the first sample $\mathbf{x}_{\tilde{l};m}$ in a class that is labeled incorrectly, i.e.,

$$y_{\tilde{l};m} \neq \tilde{y}_{\tilde{l};m} \text{ and } y_{\tilde{l};i} = \tilde{y}_{\tilde{l};i} \quad \text{for i} = 1, \ldots, \text{m} - 1 \tag{14}$$

and relabel it. Since the samples are sorted with decreasing certainty, we can expect to achieve a big correction in the model by selecting the first incorrectly labeled sample. For the next iteration, the relabeled sample and the correctly predicted samples before it in the corresponding class are added to the set of labeled samples

$$\mathcal{L} = \mathcal{L} \cup \left\{ \mathbf{x}_{\tilde{l};1}, \ldots, \mathbf{x}_{\tilde{l};m} \right\}. \tag{15}$$

This is repeated for the desired number of iterations. Since the algorithm lets the user select a sample in the wrong class in each iteration, it is called FCWL. The algorithm is summarized in Algorithm 3.

Algorithm 3. FCWL-active learning with incorrect label correction by user

```
    Input:  training points $\mathcal{X}$ and labels $\mathcal{Y}$ initial set of
            labeled samples $\mathcal{L}_0$ total number of iterations $p$
    Output: new set of labeled samples $\mathcal{L}$
Algorithm:

  for $t = 0, 1, 2, \ldots, p$ do
      Obtain $\boldsymbol{W}_t$ by training classifier on $\mathcal{L}_t$
      Compute margin $\mu_i$ for each sample according to (11)
      Predict label $\tilde{y}_i$ for each sample according to (12)
      Present samples to user according to Table II and equation (13)
      Let user relabel first sample $\mathbf{x}_{\tilde{l};m}$ with incorrect predicted
      label from one class and set $\mathcal{L}_{t+1} = \mathcal{L}_t \cup \left\{ \mathbf{x}_{\tilde{l};1}, \ldots, \mathbf{x}_{\tilde{l};m} \right\}$
  end for
  return $\mathcal{L}_t$
```

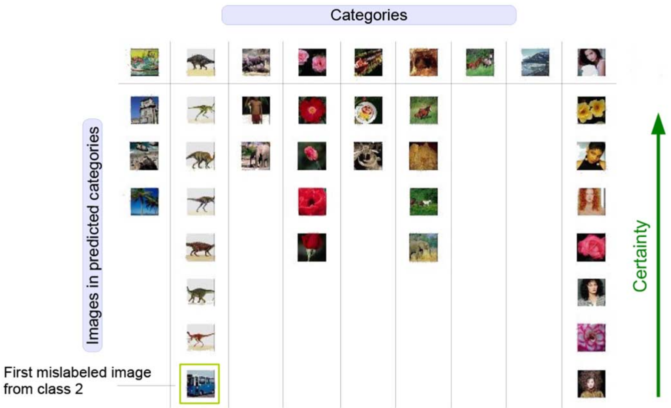

Categories
Images in predicted categories
First mislabeled image from class 2
Certainty

Fig. 3. Example list of images presented by the algorithm for an optical dataset. The first row contains images representing each class. The sample images are arranged in the columns, which correspond to the predicted class, with decreasing margin.

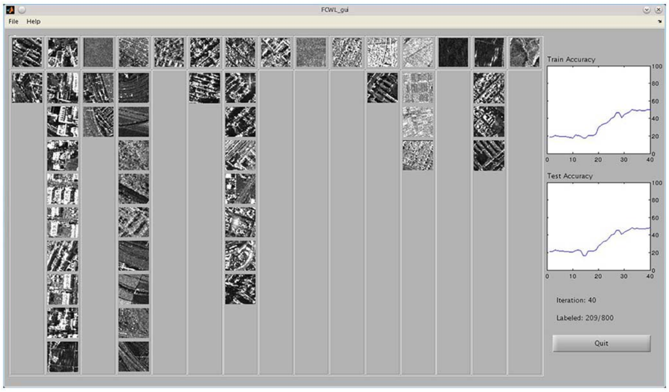
A schematic diagram of the FCWL algorithm for an optical dataset is presented in Fig. 3. On the top row, we see images representing the different categories of the dataset. Then, below them, the algorithm places the images based on their predicted labels in the corresponding categories. In this example, the algorithm has already many correct predictions in class 2 with the only incorrect prediction being the last one. So, by selecting this image the user can label all previous images from this category as correct and relabel the incorrect one.

In [2], active learning algorithms are categorized based on the sample selection heuristic, such as uncertainty sampling, expected model change, query by committee, and variance reduction. Our proposed active learning algorithm fits into the categories of uncertainty sampling and expected model change. However, the difference in which these principles are applied in this algorithm is that the samples are presented to the user in an ordered fashion and then the user selects which sample to label. Instead, in previous active learning algorithms, the sample for labeling was directly selected by the algorithm. Additionally, the expected model change of the classifier in the previous algorithms is estimated based on the gradient magnitude of the classifier optimization function for each sample. Here, the samples with the highest expected model change based on the correction introduced by the user are selected.

For the training of a classifier with the FCWL algorithm, a user interface was developed during this work. The user interface presents the list of ranked images to the user, as described in Section III, and asks the user to select the first incorrectly predicted image from one category and relabel it. Additionally, some information about the training progress is given. An example screenshot of the user interface on training with the SAR dataset is presented in Fig. 4. In this example, the training is currently at iteration 40 and 209 images have been labeled so far. Additionally, the plots show that the prediction accuracy on the test and training set is currently at about 50%.

V. Experimental Results

After introducing two novel methods for active learning, experiments are conducted to compare the introduced methods to state-of-the-art algorithms. The dataset used for evaluation is



Iteration: 40

Labeled: 209/800

Fig. 4. Screenshot of user interface for FCWL algorithm on SAR dataset after 40 iterations.

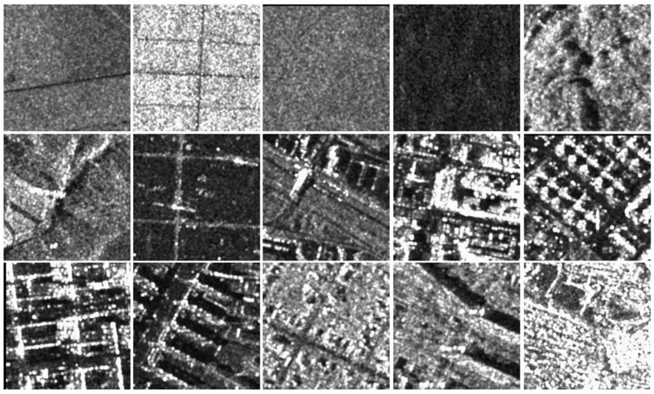


Fig. 5. Sample images of the SAR dataset.

the SAR dataset, from which different types of feature vectors are extracted. In the experiment, the accuracy of the different active learning algorithms will be presented for an increasing number of samples. Additionally, some results will be presented that show the effects of the different regularization parameters, used in the algorithms.

A. Dataset

The SAR dataset consists of a collection of 3434 images of the size 160×160 pixels, which are grouped in 15 classes consisting of various factors such as presences of forests, water, roads, and urban area density. Three different features, namely BoW model of SIFT [35], BoW model of Weber local descriptors (WLDs) [36], and Gabor [37] were extracted from the images to represent each image with a feature vector. Each feature vector is of length 64, which leads to a matrix of size 3434×64. Furthermore, the whole feature matrix is normalized to the range of $[-1, 1]$ for each experiment. Some sample images are depicted in Fig. 5.

B. Setup

In addition to our proposed method, the following active learning methods were also applied on the dataset.

1) TED [11], which defines a cost function, based on the covariance of the prediction error of a least squares classifier. Then, in each iteration, the sample which minimizes the value of the cost function, is selected for labeling.
2) MAED [12], which extends the TED algorithm with a manifold adaptive kernel, in order to incorporate the manifold structure into the selection process.
3) $\mathrm{LLR}_{\mathrm{Active}}$ [38], which minimizes the error of reconstructing the whole dataset based on the selected samples and the matrix describing the locally linear embedding.
4) $\mathrm{SVM}_{\mathrm{Active}}$ [8], which iteratively adds points closest to the boundary of an SVM classifier to the training set and trains the classifier on the new set. To extend this algorithm to multiple classes, an OVR classifier is used and the margin of each point is computed based on its distance to the corresponding winning classifier.
5) Random sampling method, which randomly selects a given number of points.

For the compared active learning algorithms, SVM with OVR scheme was used for training and classification. Linear kernel is used with the SVM. However, other kernels such as Gaussian and Chi-square were used, but the best results were achieved by using linear kernel. The metric used for comparison is the classification accuracy of the associated classifier. In order to obtain stable results, multiple tests were performed on different subsets of the dataset and the average accuracy over all subsets computed. During each experiment, a random subset was chosen from the whole dataset for training and testing. For the classifier parameter selection, we performed cross validation on each dataset, by increasing each parameter exponentially from the value 10^{-4} to to the value 10^4 and training the classifier for each parameter. Then, the parameter for which the classifier achieved the highest prediction accuracy was chosen for each dataset. Similarly, we performed cross validation for the parameter selection of the active learning algorithms. Each active learning algorithm was applied on each dataset for exponentially increasing parameter values between 10^{-4} and 10^4 and selected 10 samples for training. Then, the parameter, for which the highest prediction accuracy was achieved in each dataset, was chosen for the experiments.

C. Experiment 1: Active Learning Using TC

In this experiment, the performance of the TC-based active learning method is compared to the introduced algorithms. In addition to the proposed algorithm, which selects samples based on their distance to the TC boundary, the TC is also applied on the samples selected by the other active learning algorithms.

The experiments were repeated 10 times for each dataset and during each test a subset of 500 random samples was selected from the dataset. Then each active learning algorithm selected an increasing number of samples from 20 to 200. For classification, the samples selected by each active learning algorithm were used as a training set and all samples of the subset as a test set. The classification results are presented in Figs. 6(a)–8(a). Additionally, the stability of employed techniques is depicted in Fig. 9 as the standard deviation of accuracy over the number of trials.

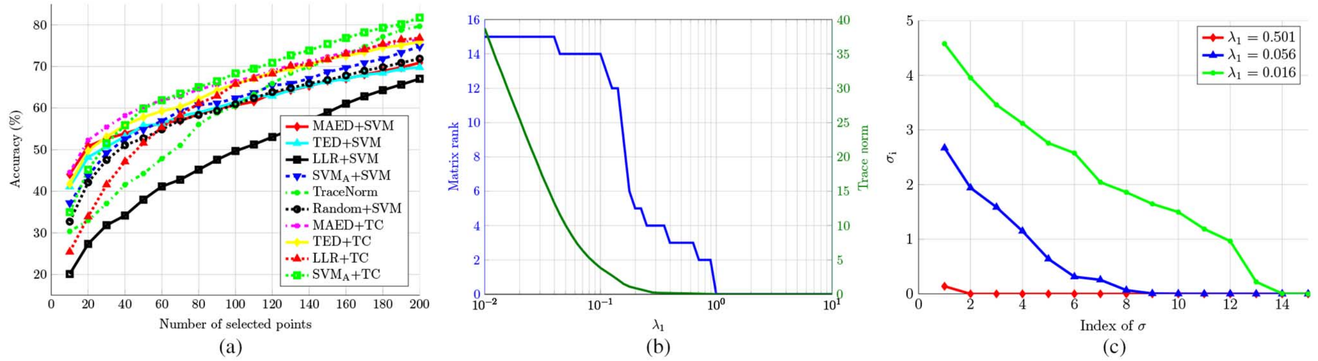


Fig. 6. Results on SAR Gabor. (a) Classification accuracy for different active learning algorithms. (b) Analysis of matrix rank and trace-norm for changing parameter λ_1. (c) Singular value spectrum for different values of λ_1.

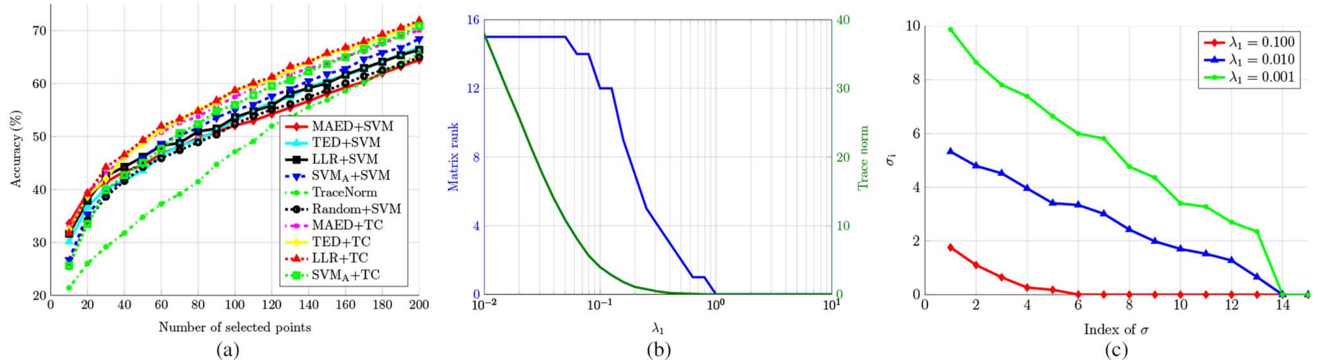


Fig. 7. Results on SAR SIFT. (a) Classification accuracy for different active learning algorithms. (b) Analysis of matrix rank and trace-norm for changing parameter λ_1. (c) Singular value spectrum for different values of λ_1.

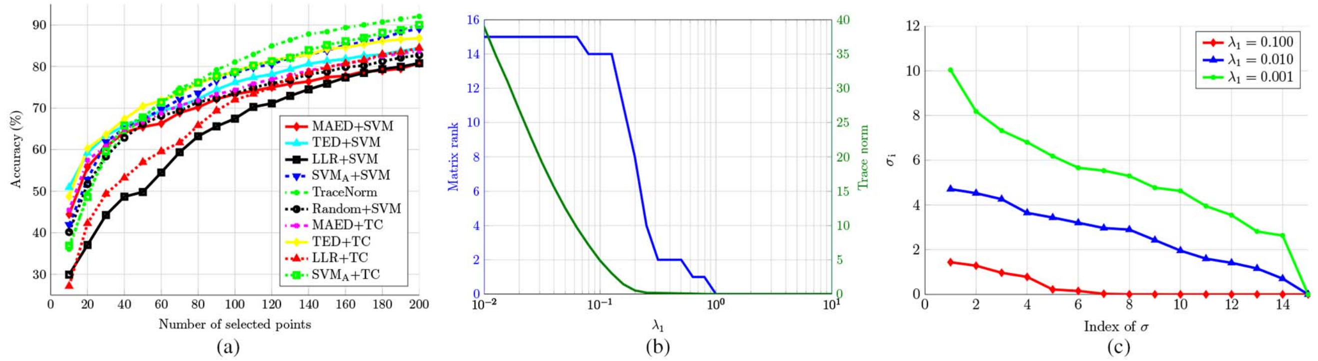


Fig. 8. Results on SAR WLD. (a) Classification accuracy for different active learning algorithms. (b) Analysis of matrix rank and trace-norm for changing parameter λ_1. (c) Singular value spectrum for different values of λ_1.

The results show that the overall accuracy of the algorithms depends on the feature descriptors chosen. Best results are achieved with the WLD feature descriptor, where some algorithms achieve an accuracy of about 90%. Next comes the Gabor feature descriptor, which has on average 10% lower accuracy and finally the SIFT feature descriptor, which again leads to 10% less average accuracy. For all active learning algorithms, we notice that coupling the algorithm with the TC nearly always leads to a higher accuracy compared to coupling the algorithm with the SVM-classifier. For the proposed active learning algorithm, based on choosing the samples based on their distance to the TC boundary, we notice that it performs poorly on the SIFT feature descriptors, but improves in performance on the Gabor feature descriptors and outperforms the other algorithms for an increasing number of samples on the WLD feature descriptors, which are the most important, since overall the highest accuracy is achieved here.

In general, we notice that algorithms based on the sample distribution, like $\mathrm{LLR}_{\mathrm{Active}}$ and MAED, perform well for a small number of samples and that algorithms based on the available label information and trained classifier perform better as the number of samples increases. This property can be explained by the fact that at the beginning the trained classifier usually lies still far away from the real classification boundary and therefore the samples it selects as informative, might actually be samples that do not contribute much information for training.

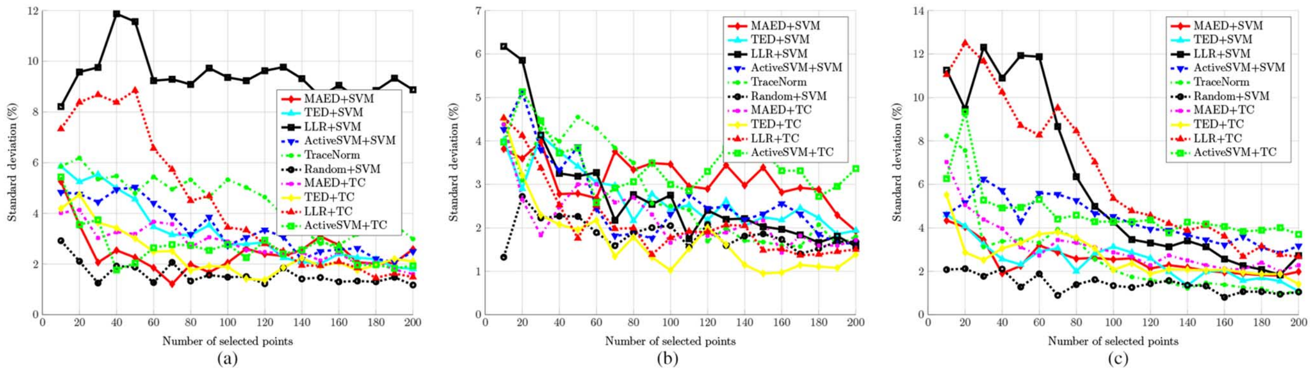


Fig. 9. Stability of different active learning algorithms on SAR dataset defined by the standard deviation of accuracy over the number of trials. (a) SAR Gabor. (b) SAR SIFT. (c) SAR WLD.

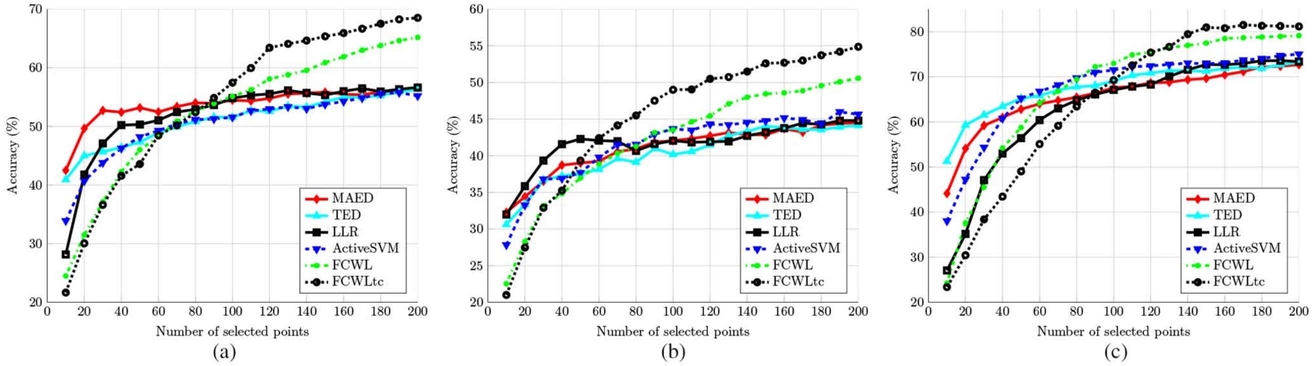


Fig. 10. Classification accuracy of different active learning algorithms on SAR dataset. (a) SAR Gabor. (b) SAR SIFT. (c) SAR WLD.

However, as the number of samples increases and the classifier finds the position of real boundary, the samples it selects for labeling are also samples that have a high probability of becoming support vectors in the next training iteration. On the other hand, algorithms that select samples based on the sample distribution, might achieve high accuracy at the beginning, since selecting those samples provides a good overall picture of how the samples are aligned in space. However, as the number of training samples increases, selecting samples in this way provides less new information, since those samples are usually located further away from the class boundary in the spatial point distribution and therefore have a lower probability of becoming support vectors.

1) TN Parameter Analysis: In order to analyze the behavior of the TC associated to the proposed active learning algorithm, experiments were also conducted on each dataset with different values of λ_1 and λ_2. For these experiments, a random subset of 1000 samples from each dataset was selected as a test set and a smaller randomly selected subset of 100 samples as a training set. Then, the TC was trained with different values of the parameters λ_1 and λ_2. The results have shown that the parameter λ_2 has less effect on the behavior and in general, the best results are achieved when λ_2 scales similarly to λ_1. Therefore, in the figures showing the behavior of the classifier with respect to the parameters, λ_2 was always chosen as $\lambda_2 = 0.1\lambda_1$.

Figs. 6(b)–8(b) show the resulting matrix rank and trace-norm of the matrix $\boldsymbol{W}$ for different values λ_1 around the area, where the matrix rank drops from the maximum rank to zero for Gabor, SIFT, and WLD features, respectively. The results show that an increase in the value of λ_1 leads to a decrease in the trace-norm and therefore the rank of the matrix from the maximum value to zero. Figs. 6(b), 7(c), and 8(c) additionally show the resulting singular values of $\boldsymbol{W}$ for different values of λ_1. Again, we see that as the value of λ_1 increases, the average value of the singular values decreases and more singular values become zero.

D. Experiment 2: Visualization-Based Active Learning

Similar experiments were performed for the analysis of the interactive visualization-based active learning. The proposed active learning algorithm was applied in conjunction with the TC and SVM classifier, while for the other algorithms, the SVM classifier was trained on the selected points. The experiments were repeated again 10 times for each dataset, but here in order to keep the results objective due to the multiple point selection of the proposed active learning classifier, different subsets were used for training and for testing. Specifically, during each experiment, a subset of 500 samples was selected as a training set and a different subset of 500 samples as a test set. Making the test set completely separated from training set leads to a reduction in accuracy for all algorithms. The classification results are presented for the three datasets in Fig. 10(a)–(c). Additionally, the stability of algorithms is depicted in Fig. 11. Again, we see a dependence of the overall accuracy on the choice of feature descriptors with WLD leading to the highest

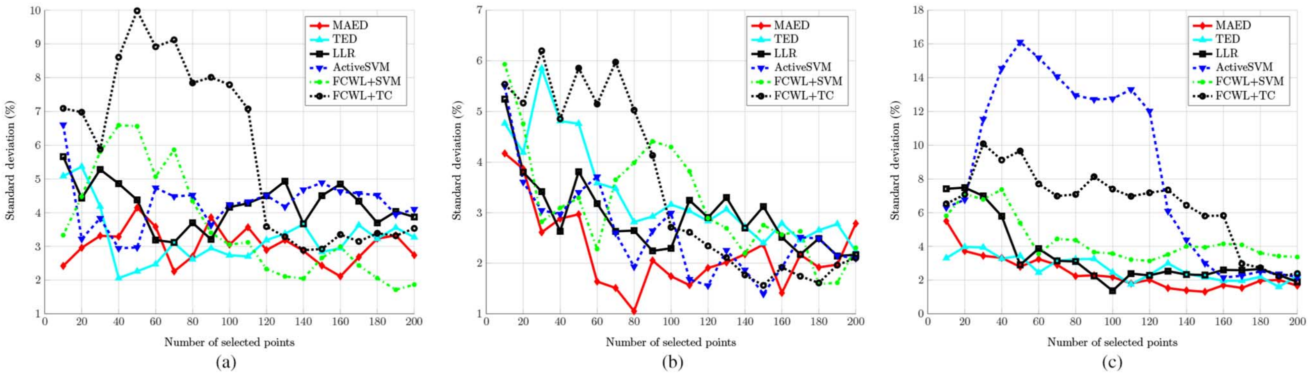


Fig. 11. Stability of different active learning algorithms on SAR dataset defined as the standard deviation of accuracy over the number of trials. (a) SAR Gabor. (b) SAR SIFT. (c) SAR WLD.

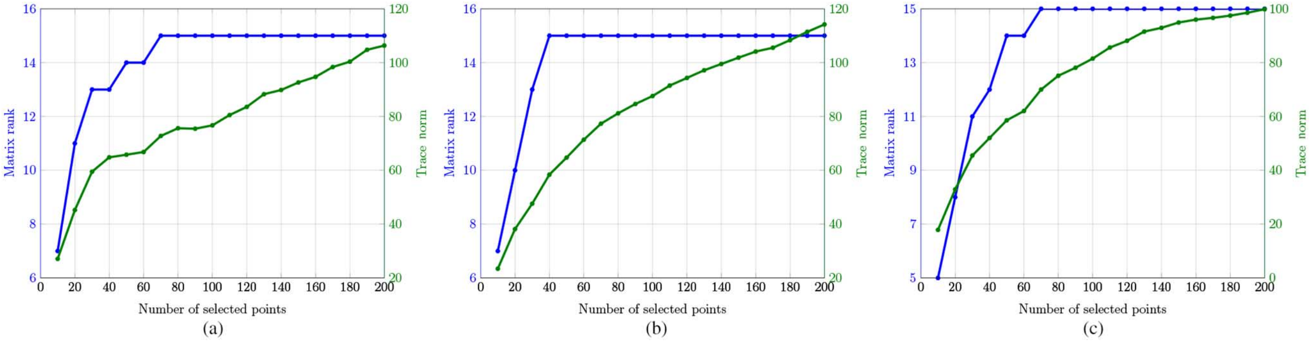
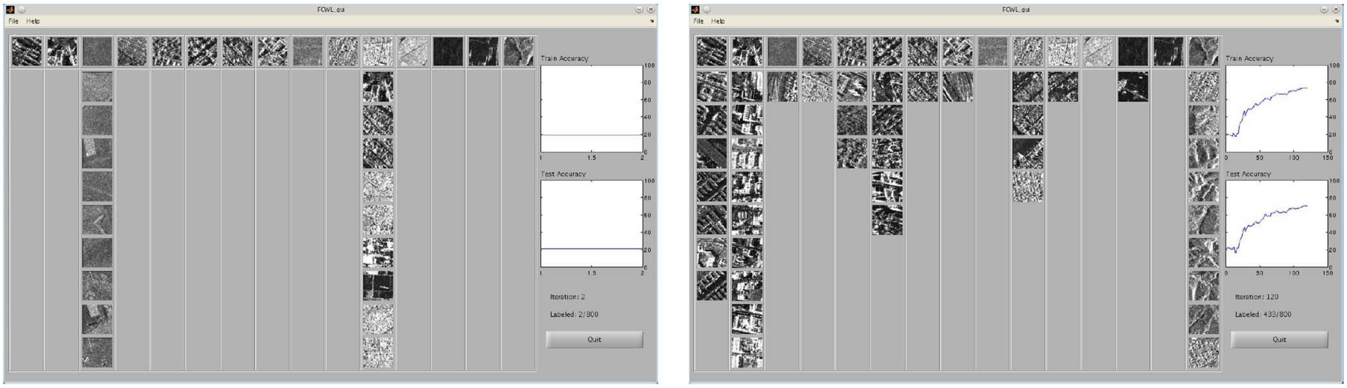


Fig. 12. Matrix rank and trace-norm of FCWL algorithm with TC on SAR dataset. (a) SAR Gabor. (b) SAR SIFT. (c) SAR WLD.

accuracy and SIFT to the lowest. For the FCWL algorithm, the plots show that for both SVM and TC it outperforms the other active learning algorithms with all feature descriptors for an increasing number of interactions, with the TC doing best in the end. The improved performance of the FCWL algorithm with TC compared to FCWL with SVM can be explained again by the ability of the TC to deal with a high dimension of the feature space and a high number of samples. This gets amplified even more in the case of FCWL, where multiple samples are selected per iteration, leading to an overall higher number of samples. Thus, the FCWL with TC can repeatedly make more accurate predictions, which lead to even more samples being labeled correctly and thus an even higher performance after training. This property becomes clear in the plots, where the FCWL with TC significantly outperforms the other algorithms, as the number of selected samples increases beyond 200. However, the effect of the increasing speed of performance improvement due to the increased accuracy and therefore higher amount of correctly predicted labels, can also be observed with the FCWL and SVM classifier, where we see again that for 200 samples it has a high difference in performance, compared to the other algorithms.

For a small number of samples, we notice that the FCWL algorithm does worse than the other algorithms. The reason for this can be found as before in the difference between classifier based on sample distribution-based algorithms. When the number of samples is small, the predictions of the FCWL are still inaccurate and therefore the samples labeled by the user might not be the most informative. Additionally, it is possible that at the initial steps, the FCWL algorithms know only a part of the classes and therefore do not predict some classes at all, which can lead to a further decrease in accuracy. However, as the number of samples increases, all classes can be predicted and the overall accuracy increases faster. On the other hand, algorithms like MAED and TED perform well at the beginning due to the selection of a more diverse set of samples, which usually contains all classes, but keep increasing slower in accuracy later, as a high sample diversity is no longer coupled to a high increase in classifier performance at this point.

Fig. 12(a)–(c) shows the evolution of the matrix rank and trace-norm for the FCWL algorithm with trace-norm classifier and an increasing number of samples. These plots make the behavior of the FCWL algorithm with the trace-norm classifier more clear. At the beginning we see a low matrix rank, which means that only a small subset of classes are known. The number of samples required, until all classes are identified varies for each feature descriptor. For example, for the Gabor descriptor all classes are known after 80 samples have been selected, for the SIFT descriptor 40 samples are necessary and for the WLD descriptor 70. However, even at this point some of the new classes might still be represented only weakly. Therefore, the accuracy plots show that the FCWL algorithm with trace-norm classifier achieve good performance, after the number of samples is beyond the point, where all classes have been identified. The plots of the trace-norm value in Fig. 12(a)–(c) show how the trace-norm behaves as a relaxation of the matrix rank. It keeps increasing fast at the beginning as the rank of the matrix is growing and has a lower slope in the end when the rank of the matrix is constant.



Iteration: 2

Labeled: 2/800

Iteration: 120

Labeled: 433/800

(a) (b)

Fig. 13. Screenshots of user interface for FCWL algorithm on SAR dataset. (a) After two iterations. (b) After 120 iterations.

VI. Conclusion

We have introduced a novel active learning algorithm that with an increasing number of samples to label, the effects of overfitting can be reduced by minimizing the rank of the matrix. This property is confirmed by the experiments, where the proposed algorithm achieves the highest accuracy for an increasing number of samples. However, a disadvantage of the algorithm is the increasing computational effort required to solve the objective with a coordinate descend algorithm. Developing more efficient methods to solve the optimization problem is therefore one possible direction for future work. Additionally, as the experimental results show, for a small number of selected samples, the proposed algorithm is outperformed by algorithms, which select points based on the sample distribution, e.g., MAED. Therefore, another possible direction for future work is to combine the two approaches in order to develop an algorithm that takes sample distribution and label information in account and consistently provides high accuracy.

Appendix A
User Interface Screenshots

Fig. 13 shows two screenshots of the FCWL algorithm user interface on the SAR WLD dataset, to illustrate the training process. In Fig. 13(a) after two iterations only two classes of the dataset have been learned and therefore, all images are predicted in these two classes. However, as the training progresses the predictions become more diverse and overall more images get predicted correctly. This can be seen in Fig. 13(b) after 120 iterations, where the predictions are more correctly distributed over all classes.

References

[1] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Boston, MA, USA: Houghton Mifflin Harcourt, 2013.

[2] B. Settles, "Active learning literature survey," Univ. Wisconsin, Madison, WI, USA, Computer Sciences Tech. Rep. 1648, 2010.

[3] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 606–617, Jun. 2011.

[4] B. Settles, "Active learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 6, no. 1, pp. 1–114, 2012.

[5] C. Persello *et al.*, "Cost-sensitive active learning with lookahead: Optimizing field surveys for remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6652–6664, Oct. 2014.

[6] L. B. Patra, "A novel som-svm based active learning technique for image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6899–6910, Nov. 2014.

[7] Z. Harchaoui, M. Douze, M. Paulin, M. Dudik, and J. Malick, "Large-scale image classification with trace-norm regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR'12)*, Jun. 2012, pp. 3386–3393.

[8] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, pp. 107–118.

[9] M. Wang and X.-S. Hua, "Active learning in multimedia annotation and retrieval: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, p. 10, 2011.

[10] A. C. Atkinson, A. N. Donev, and R. D. Tobias, *Optimum Experimental Designs, With SAS*. New York, NY, USA: Oxford Univ. Press, 2007.

[11] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proc. 23rd Int. Conf. Mach. Learn.* New York, NY, USA: ACM, 2006, pp. 1081–1088.

[12] D. Cai and X. He, "Manifold adaptive experimental design for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 707–719, Apr. 2012.

[13] N. Panda, K.-S. Goh, and E. Y. Chang, "Active learning in very large databases," *Multimedia Tools Appl.*, vol. 31, no. 3, pp. 249–267, 2006.

[14] S. Vijayanarasimhan and K. Grauman, "What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR'09)*, 2009, pp. 2262–2269.

[15] L. Bao, J. Cao, T. Xia, Y.-D. Zhang, and J. Li, "Locally non-negative linear structure learning for interactive image retrieval," in *Proc. 17th ACM Int. Conf. Multimedia*, 2009, pp. 557–560.

[16] R. Yan, L. Yang, and A. Hauptmann, "Automatically labeling video data using multi-class active learning," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 516–523.

[17] L. Zhang *et al.*, "Active learning based on locally linear reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2026–2038, Oct. 2011.

[18] S. Patra and L. Bruzzone, "A batch-mode active learning technique based on multiple uncertainty for SVM classifier," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 497–501, May 2012.

[19] B. Geng, Y. Yang, Z.-J. Zha, C. Xu, and X.-S. Hua, "Unbiased active learning for image retrieval," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2008, pp. 1325–1328.

[20] M. Wang *et al.*, "Interactive video annotation by multi-concept multi-modality active learning," *Int. J. Semant. Comput.*, vol. 1, no. 4, pp. 459–477, 2007.

[21] H. Sahbi, P. Etyngier, J.-Y. Audibert, and R. Keriven, "Manifold learning using robust graph laplacian for interactive image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR'08)*, 2008, pp. 1–8.

[22] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[23] V. Vapnik, *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. New York, NY, USA: Springer, 1982.

[24] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *Proc. 17th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 1994, pp. 3–12.

[25] R. A. Redner and H. F. Walker, “Mixture densities, maximum likelihood and the EM algorithm,” *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.

[26] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, “A maximum entropy approach to natural language processing,” *Comput. Ling.*, vol. 22, no. 1, pp. 39–71, 1996.

[27] X. Zhu, J. Lafferty, and Z. Ghahramani, “Combining active learning and semi-supervised learning using gaussian fields and harmonic functions,” in *Proc. ICML 2003 Workshop Continuum From Labeled to Unlabeled Data in Mach. Learn. Data Mining*, 2003, pp. 58–65.

[28] B. Demir and L. Bruzzone, “A multiple criteria active learning method for support vector regression,” *Pattern Recognit.*, vol. 47, no. 7, pp. 2558–2567, 2014.

[29] B. Demir, L. Minello, and L. Bruzzone, “An effective strategy to reduce the labeling cost in the definition of training sets by active learning,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 79–83, Jan. 2014.

[30] S. Patra and L. Bruzzone, “A cluster-assumption based batch mode active learning technique,” *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1042–1048, 2012.

[31] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Dec. 2004.

[32] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines And Other Kernel-based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[33] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, May 2011.

[34] K. Chen, *Matrix Preconditioning Techniques and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[35] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

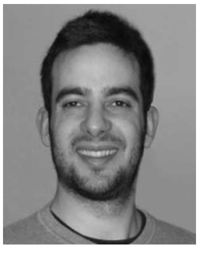
[36] J. Chen *et al.*, “A robust descriptor based on weber’s law,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR’08)*, 2008, pp. 1–7.

[37] C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition,” *Proc. IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, 2002.

[38] L. Zhang *et al.*, “Active learning based on locally linear reconstruction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2026–2038, Oct. 2011.

Mohammadreza Babaee (S’13) received the B.S. degree in computer engineering from the University of Isfahan, Isfahan, Iran, in 2004, and the M.S. degree in biomedical computing from Technische Universität München, Munich, Germany, in 2012. Currently, he is pursuing the Ph.D. degree in electrical engineering at the Technische Universität München.

He joined the Image Processing and Underwater Vision Lab, University of Miami, Coral Gables, FL, USA, as a Visiting Scholar to accomplish his master thesis. His research interests include computer vision, data mining, and multimedia.



Stefanos Tsoukalas received the B.S. and M.S. degree. in mechanical engineering from the Technische Universität München, Munich, Germany, in 2012 and 2014, respectively.

His research interests include numerical computation, computer vision, and machine learning.

Gerhard Rigoll (SM’98) was born in Essen, Germany, in 1958. He received the Dipl.-Ing. degree in technical cybernetics and the Dr.-Ing. degree in automatic speech recognition from Stuttgart University, Stuttgart, Germany, in 1982 and 1986, respectively.

After working as a Researcher in the USA and Japan for several years, he was appointed as a Full Professor of Computer Science with Gerhard Mercator University, Duisburg, Germany, in 1993. In 2002, he joined Technische Universität München, Munich, Germany, heading the Institute for Human–Machine Communication. He is the author and co-author of more than 500 papers in his research fields. His research interests include human–machine communication, multimedia information processing, speech, handwriting and gesture recognition, face detection and identification, emotion recognition, person tracking, information retrieval, video-indexing, and interactive computer graphics.

Mihai Datcu (F’13) received the M.S. and Ph.D. degrees in electronics and telecommunications from the University Politechnica of Bucharest (UPB), Bucharest, Romania, in 1978 and 1986, respectively, and the title *Habilitation á diriger des recherches* in computer science from the University Louis Pasteur, Strasbourg, France, in 1999.

Since 1981, he has been a Professor with the Department of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology (ETTI), UPB, working on image processing and electronic speckle interferometry. Since 1993, he has been a Scientist with the German Aerospace Center (DLR), Oberpfaffenhofen, Germany. He is developing algorithms for model-based information retrieval from high complexity signals and methods for scene understanding from very high-resolution synthetic aperture radar (SAR) and interferometric SAR data. He is engaged in research related to information theoretical aspects and semantic representations in advanced communication systems. Currently, he is a Senior Scientist and an Image Analysis Research Group Leader with the Remote Sensing Technology Institute (IMF) of DLR. Since 2011, he has also been leading the Immersive Visual Information Mining Research Laboratory, Munich Aerospace Faculty, Munich, Germany, and he is the Director of the Research Center for Spatial information at UPB. He has held Visiting Professor appointments with the University of Oviedo, Oviedo, Spain; the University Louis Pasteur, Strasbourg, France; the International Space University, Strasbourg, France; University of Siegen, Siegen, Germany; University of Innsbruk, Austria; University of Alcala, Spain; University Tor Vergata, Rome, Italy; Universidad Pontificia de Salamanca, campus de Madrid, Spain; University of Camerino, Camerino, Italy; and the Swiss Center for Scientific Computing (CSCS), Manno, Switzerland. From 1992 to 2002, he had a longer Invited Professor assignment with the Swiss Federal Institute of Technology, Eidgenoessische Technische Hochschule Zurich, Zurich, Switzerland. Since 2001, he has initiated and led the Competence Centre on Information Extraction and Image Understanding for Earth Observation at ParisTech, Paris Institute of Technology, Telecom ParisTech, Paris, France, a collaboration of DLR with the French Space Agency (CNES). He has been a Professor of the DLR-CNES Chair at ParisTech, Paris Institute of Technology, Telecom ParisTech. He initiated the European frame of projects for image information mining (IIM) and is involved in research programs for information extraction, data mining and knowledge discovery, and data understanding with the European Space Agency (EPA), NASA, and in a variety of national and European projects. He is a Member of the European Image Information Mining Coordination Group (IIMCG). He and his team has developed and are currently developing the operational IIM processor in the Payload Ground Segment systems for the German missions TerraSAR-X, TanDEM-X, and the ESA Sentinel 1 and 2. He is the author of more than 200 scientific publications, among them about 50 journal papers, and a book on number theory. His research interests include Bayesian inference, information and complexity theory, stochastic processes, model-based scene understanding, image information mining, for applications in information retrieval and understanding of high-resolution SAR and optical observations.

Dr. Datcu has served as a Co-Organizer of international conferences and workshops, and as Guest Editor of special issue on IIM of the IEEE and other journals. He was the recipient of the Best Paper Award in 2006, IEEE Geoscience and Remote Sensing Society Prize, the National Order of Merit with the rank of Knight in 2008, for outstanding international research results, awarded by the President of Romania, and the Romanian Academy Prize Traian Vuia for the development of SAADI image analysis system and activity in image processing in 1987.