Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Kartographie

Approaching a collective place definition from street-level images using deep learning methods

Hao Lyu

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.

Vorsitzender:	Prof. Dr.-Ing. Uwe Stilla
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Liqiu Meng
	2. Prof. Dr. Wolfgang Kainz

Die Dissertation wurde am 05.12.2018 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 13.02.2019 angenommen.

Contents

Zusammenfassung

Sowohl im täglichen Leben als auch in vielen Forschungsbereichen ist die Beantwortung der “Wo” - Frage eine wesentliche Aufgabe. Dabei handelt es sich oft um die Ortsidentifizierung und nicht nur um eine Positionierungsaktion. Ort ist ein Begriff, der reichliche semantische Information enthlt, und innerhalb eines spezifischen geographischen Raumes begrenzt ist. Eine der einfachsten Möglichkeiten, einen Ort wahrzunehmen und seine Information mit anderen zu teilen, ist es, Bilder davon zu machen. Bilder enthalten oft semantische sowie rumliche Eigenschaften in verschiedenen Detaillierungsgraden. Ein Bild von einer sichtbaren Szene wie einer Wste oder einem Strand kann uns helfen, das Bild in einem relativ groben Detaillierungsgrad zu lokalisieren. Dagegen liefert ein Bild, das materielle Gegenstände wie Gebäude, Straenschilder, oder topologische Merkmale eines Geländes enthält, starke Hinweise darauf, die Geo-Lokalisierung in einem relativ feinen Detaillierungsgrad zu untersttzen. Die rumliche Eigenschaft umfasst einerseits die räumliche Stelle der materiellen Gegenstände an einem Ort; andererseits spiegelt sie die Stelle der Kamera, von der die materiellen Gegenstände fotografiert werden. Die radikale Entwicklung von Deep Learning verbessert das maschinelle Verständnis von Bildinhalten. In dieser Arbeit wird die Herausforderung angesprochen, einen Ort zu verstehen, indem die visuellen und rumlichen Eigenschaften in freiwillig erhobenen Bilder auf Straenebene mit komparativen Lernen und Variationsautokodierer (VAE).

In dieser Arbeit werden einen Rahmen vor, um die Ortsdefinition mit einer Datenperspektive zu betrachten. Dementsprechend werden zwei Orten Darstellungen aus dem Latentes Variablenmodell vorgeschlagen. Beide Darstellungen kodieren visuelle Inhalte und räumliche oder semantische Eigenschaften aus Bildbeobachtungen über Orte. Eine, die implizit kategorialen oder rumlichen Informationen codiert, repräsentiert Orte mit einer einzelnen Wahrscheinlichkeitsverteilung oder einer Mischverteilung. Die andere beschftigt sich mit einer holistischen latenten Ortsdarstellung, die die Kameraposition und die visuellen Eigenschaften voneinander abgrenzt. Bei einer Kamerapose und einer entsprechenden Ortsdarstellung kann das Modell ein Bild erzeugen, von dem angenommen wird, dass es von der gegebenen Kamerapose aufgenommen wird.

Es werden drei Experimente durchgefhrt, um die vorgeschlagenen Ideen ber die visuellen und rumlichen Informationen in probabilistischen Ortsdarstellungen kombiniert zu überprüfen. Im ersten Experiment verwenden wir Triplet-VAE und kategoriale In-

formationen, d. H. die Kennzeichnung verschiedener Orte verbindet verschiedene Orte mit einer einzigen Normalverteilung. Das Ergebnis zeigt, dass die latente Ortsdarstellung kategoriale Informationen zu Orten kodiert. Im zweiten Experiment werden eine gaussian mixture Ortsdarstellung (GMPR) vorgeschlagen. Eine Kombination aus vergleichenden Lernmodellen und VAE wird verwendet, um eine solche Ortsdarstellung zu erlernen. Bei vordefinierten Ortsinformationen wird die Center-Triplet-VAE-Modellen zum Erlernen des GMPR verwendet. Wenn keine vordefinierten Ortsinformationen vorhanden sind, werden die Positionen (aus Kameraposition) der Bildbeobachtungen als schwaches Uberwachungssignal verwendet, um das Lernen zu untersttzen. Das gelernte GMPR bewahrt einerseits die visuellen und rumlichen Informationen, um neuartige Bilder ber einen bestimmten Ort zu erzeugen, andererseits kann es zu Unterscheidung die Herkunft das bestimmtes Bilder verwendet werden. Das dritte Experiment testet die ganzheitliche Darstellung eines latenten Ortes. Mit gegebenen Ortsdarstellung und einer Kamerapose kann das Modell ein entsprechendes Bild erzeugen.

Obwohl das Verstehen der visuellen und rämlichen Eigenschaften nur der erste Schritt ist, um der Begriff des Ortes anzugehen, kann die erlernte Ortsdarstellung als Beweis oder Hypothese verwendet werden, um weitere Deep-Learning-Aufgaben zu untersttzen, die komplexeren rumlichen intelligenten Problemen gewidmet sind.

Abstraction

In both daily life and many research fields, answering the "where" question is one of the essential tasks. It often deals with place identification rather than just a positioning action. Place is a concept of rich semantic information bounded within a specific geographic space. One of the most straightforward ways of perceiving a place and sharing its information with others is to take pictures of it. Images often contain both semantic and spatial properties at various levels of details. An image of a visible scene such as a desert or a beach may help us to localize the image at a relatively coarse level of detail. An image containing tangible objects, such as buildings, street signs, topographic features, provides strong clues to support geo-localization at a relatively fine level of detail. The spatial property, on the one hand, includes the spatial location of tangible objects in a place, on the other hand, reflects the camera's location from which the tangible objects are photographed. The radical development of deep learning improves machine understanding of image contents. In this work, the author addresses the challenge of understanding a place by detecting visual and spatial property from voluntarily collected images with a combination of comparative learning methods and variational autoencoder (VAE).

The author proposes a framework to approach place definition with a data-driven perspective. Accordingly, two place representations are proposed from the latent variable model. Both representations encode visual contents and spatial/semantic properties from image observations about places. The one that implicitly encodes categorical or spatial information represents places with a single probabilistic distribution or a mixture of multiple distributions. Such place representation can be learned by combining comparative learning methods with VAE. The other deals with a holistic latent place representation, which disentangles the camera pose and visual properties. In this place representation, camera pose is used as a condition in VAE. Given a camera pose and corresponding place representation, the model is able to generate an image which is assumed to be taken by the given camera pose.

Three experiments are carried out to verify the proposed ideas of combining visual and spatial information in probabilistic place representations. In the first experiment we use Triplet-VAE and categorical information, i.e., the label of different places connects different places with a single normal distribution. The result shows that the latent place representation encodes categorical information about places. In the second experiment, a Gaussian

Mixture Place Representation (GMPR) is proposed. A combination of comparative learning methods and VAE is used to learn such place representation. With pre-defined place information, we use the Center-Triplet-VAE method to learn the GMPR. When no pre-defined place information, locations (from camera pose) of image observations are used as weakly supervision signal to help the learning. The learned GMPR, on the one hand, preserves the visual and spatial information to generate novel images about a given place, on the other hand, can be used to distinguish from which place a given image comes. The third experiment tests the holistic latent place representation. Given the place representation and a camera pose, the model is able to generate a corresponding image.

Understanding the visual and spatial properties is just the first step to approach the concept about a place, the learned place representation can be used as evidence or hypothesis to support further deep learning tasks dedicated to more complex spatial intelligent problems.

Chapter 1

Introduction

1.1 Background

1.1.1 Locations and Places - the Science of "where"

In daily life, proposing and solving "where" problem compose a large portion of our daily activities from the morning rush hour to the evening leisure time. Some of the decisions are too familiar to be noticed as we make them too often in our life as a daily routine, e.g., how to get from the home to the workplace and return. However, people may still take some thoughts to answer other "where" questions, such as asking for a meeting place with a friend. The answer to the question "Where shall we meet later?" can take various forms. It can be a place where both the two persons are familiar with or a message with location description from mobile phone applications. The meeting place can be described as, for example, in an underground station, in front of a cafe, a pinpoint on Google maps, or even merely coordinate values of a GPS reading.

Researchers are also interested in the "where" question, where spatial knowledge is concerned. Spatial knowledge is the knowledge about geographic environments that enable human or animals to perform spatial tasks, e.g., navigation and wayfinding (Montello, 2005). Researchers from different subject fields have various perspectives on spatial knowledge. In this work, we name a few among them as examples. In cognition and behavior research, it is important to understand the mechanism behind spatial cognition, i.e., acquisition, organization, utilization, and revision of spatial knowledge. Many neuroscientists are interested in identifying the functional regions in the neural system that support the cognition of spatial environments. Scientists and engineers from robotics are making efforts to enable robots and other autonomous machines to perform spatial tasks. Researchers have designed various spatial cognition mechanisms for autonomous machines to support them doing these tasks. In geographic information science (GIS) and cartography, spatial knowledge consists of entities in a geographic environment, their spatial locations, and the relationship between them. One of the major tasks in GIS and cartography is to survey, record, organize, and represent spatial knowledge.

Spatial knowledge as a cognitive map

In behavior geography and psychology, "cognitive map" is a widely accepted hypothesis about spatial knowledge. Cognitive map, first proposed by Tolman (1948), is a hypothesis of spatial knowledge for rats, and by analogy, for humans. In his experiments, the animals act as if they had a map within their neural system to guide their movements. In the most general terms, a cognitive map is a mental construct which we use to understand and know the environment (Kaplan, 1973). The evolving of the term "cognitive map" and "cognitive mapping" is reviewed comprehensively by Kitchin (1994). From a geographers perspective, Golledge (1999) proposed a four elements model to represent spatial knowledge in terms of spatial distributions of places and features. The four elements are *points*, *lines*, *areas*, and *surface*. Lynch (1960) associated urban structures with a cognitive map in cities based on a study in several American cities. He summarized five urban structures as cognitive map elements, i.e., *path*, *edges*, *districts*, *nodes*, and *landmarks*. *Landmark*, *Route*, *Survey\Configuration* model (LRS) is probably the most widely accepted spatial knowledge representation system (Siegel & White, 1975; Thorndyke & Goldin, 1983). Couclelis, Golledge, and their colleagues proposed an anchor point hypothesis to model spatial knowledge organization in a decision making context (Couclelis et al., 1987; Golledge, 1978). Discrete places are the basic elements in the anchor point hypothesis. In this hypothesis, a place is a region with a spatial extent and some salient spatial or semantic properties. Places are "anchored" in geographic space through landmarks. Landmarks are a strong spatial cue with salient characters that help people to recognize and organize geographic places. Visually singularity or sharp contrast with the surroundings is an important character in identifying landmarks (Sorrows & Hirtle, 1999).

Functional neural cells reflect spatial knowledge

Decades of neurological studies support the hypothesis of "cognitive map". The evidence has been accumulated since the early efforts from O'Keefe and Nadel (O'Keefe & Nadel, 1978). The hippocampus, a brain region of long-term memory, is suggested to store spatial knowledge and serve as a cognitive map. A collection of comprehensive reviews covers a broad range of progress under this topic (Burgess, 2008; T. Hartley et al., 2013; Moser et al., 2008, 2017). "Places" in the neural system are characterized by the activation of *place cells*, while "locations" can be estimated from a collaborative firing pattern among *grid cells*. These cells, together with other spatial related cells, such as head-direction cells which encode spatial knowledge to support human and animal's navigation and wayfinding behavior. In the following, we briefly introduce three major cell types to sketch a representation of spatial knowledge in the brain.

- **Place cell** bursts firing when an animal is within a certain region (also known as "place field") while keeps almost silent at most other regions in the environment (O'Keefe & Conway, 1978; O'Keefe & Dostrovsky, 1971; O'Keefe & Nadel, 1978). Each place cell fires at different locations, such that a group of neighboring place

cells in the hippocampus represent the entire accessible environment (Wilson & McNaughton, 1993).

- **Grid cell** fires at multiple locations when the animal is moving in the environment, despite constant changes in humans or animals running speed and running direction (Boccara et al., 2010; Doeller et al., 2010; Hafting et al., 2005). The firing pattern of the individual grid cell is associated with a hexagonal or triangular tessellation covering the entire space accessible to the animal. The observations indicate that grid cells are topologically well organized, forming a multi-scale metric representation from coarse to fine.

- **Boundary cell** is characterized as firing along boundaries of a local environment (including vertical surfaces and drop edges) within a certain distance (T. Hartley et al., 2000; Lever et al., 2002; O'Keefe & Burgess, 1996). The property of boundary cells points out the importance of geometric information of a local environment in defining the location of firing in place cells and grid cells. However, their function in the animals perception of self-location remains unclear.

The formation of a place cell is relevant to various inputs, such as grid-cells, the sense of environmental geometry, a combination of grid cells and boundary vector cells (Barry et al., 2006). Although self-motion cues along can activate a place cell, changing the appearance of the environment also activates place cell indicating the importance of visual input (O'Keefe, 1976). Experiments on rats show that the place cell will update to the correct location according to the external visual landmarks.

Mapping spatial knowledge for robots

Designing spatial cognition mechanisms to support the robot's navigation is a long-standing research topic in the name of robotic mapping (Thrun, 2003). Place recognition and localization are two basic demands in robotic navigation. Simultaneous localization and mapping techniques (SLAM) address both problems at the same time (Leonard & Durrant-Whyte, 1991). Place recognition is particularly important in SLAM, such that a robot knows whether the current place is previously mapped or a new one to be mapped such that it "closes the loop" and avoids an ever-increasing map (Newman & Kin Ho, 2005).

Spatial knowledge modeling for robotic mapping, on the one hand, is largely influenced by the research of spatial cognition, on the other hand, is deeply entangled with the development of sensors and data processing technology. In the 1970s, Kuipers (1978) proposed a TOUR model of a street network environment to support robots solving spatial tasks, i.e., route planning and relative position describing via rule-based inference technics. Schölkopf and Mallot (1995) introduced the notion of *view graph* as a discrete representation covering the whole environment for the robots navigation and path planning. Werner et al. (2000)

proposed a *route graph* concept for merging possible routes for robots, which is an analog to animals and humans spatial knowledge acquirement during navigation. These models and their followers share many similarities with spatial cognition models. In these graph-based representations, nodes contain spatial and semantic information (e.g., landmarks, fingerprints, and keyframes) that help robots localization and decision-making during navigation, while edges indicate the relationships (e.g., connectivity or routes) between two nodes. *Occupancy-grid* and *boundary model* are two metric based spatial models that are closer to the sensory data. Occupancy-grids represent occupied and free space of an environment by decomposing space into uniform rectangular (for 2D) or cubic (for 3D) cells and storing for each cell whether it is (at least partially) occupied or (entirely) free (Elfes, 1989; Moravec, 1996; Moravec & Elfes, 1985). Boundary models represent spatial knowledge by recording the boundaries of objects with geometric primitives (e.g., lines, points), and thus distinguish between occupied space and free space (e.g., Y. Liu et al. 2001; Surmann et al. 2003). It is worth pointing out that there are also landmarks in robotic navigation in terms of different sensory input, and thus are very different from the landmarks in human navigation (e.g., Dissanayake et al. 2001; Guivant et al. 2002; Guivant and Nebot 2001).

Position, location, and place in spatial knowledge

The interpretations of spatial knowledge may be varying from discipline to discipline. However, they share three fundamental concepts, i.e., position, location, and place. Both position and location are often used to refer specific spatial point in a given (local or global) coordinate systems. In this sense, positioning and localization are interchangeable when we refer "to determine a spatial point in a given coordinate system". In some cases, location and position are distinguished with subtle distinction. Location implies more on the geometric property while positions bear more topological meaning about relations. One consequence of this usage is that locations are geometrically equal, but positions are unequal when they are nodes in graph-based models or hierarchical structures. Location and position can be used to depict the spatial property of an entity including its geographic location and its relations to other entities. "Place" is a cognitive concept that enables mentally structuring of the spatial aspects of reality (Bennett & Agarwal, 2007). Place reflects the way people perceive, understand, and interact with their environment (Tuan, 1977). It attracts the attention of many geographers as the study of place is to find "a way of understanding the world" (Cresswell, 2014, p. 11). Given the importance of place in science and social life, it is difficult to formalize the concept of place without ambiguous due to its vagueness. The connotation and extension of place can be very broad as it "performs a variety of functions in different settings" depending on the background and the purpose of the analysis (Goodchild, 2011). For simplicity, some researchers reduce a place to a geographic setting acting as a spatial container of human activities and environmental entities.

1.1.2 Bringing Places to Location-based Services

Location-based Service (LBS) researches aim at deriving, modeling, communicating, and analyzing of location-based information. Locations delivered by global navigation satellite systems (GNSS), especially the global positioning system (GPS), triggered the research and development of LBS in the early 1990s. LBS have been rapidly developing since the early 2000s, after the removal of selective ability from GPS by the US President Bill Clinton. In recent years there are more and more diverse LBS applications running on various mobile devices, the attributes and semantic information attached to locations are getting more attention (Huang et al., 2018). The combination of location and platial information enriches the contents that LBS applications can provide. Examples include landmark-based navigations, location-based recommendations, and health monitoring (Adibi, 2015; Duckham et al., 2010; Huang, 2016). Analyzing the attributes and semantic information from LBS-generated data facilitates a better understanding of humans subjective attitude and social activities (e.g., Gebru et al. 2017).

GNSS embedded navigation guidance (e.g., car navigation and pedestrian navigation system) is one of the most widely used LBS applications (Raper2007a). It is equipped in vehicles to help drivers wayfinding and navigation. This commercially mature and successful application faces new challenges in the emerging navigation scenarios, e.g., indoor navigation and autonomous driving. Indoor navigation often suffers from the lack of reliable positioning methods and geo-databases (Raper et al., 2007a, 2007b). Different location sensor technologies, including Wi-Fi, RFID, Bluetooth, and digital camera have been tested and deployed for indoor positioning (Davidson & Piche, 2017). Efforts have been made to turn floor plans and 3D building models into indoor geo-databases as well.

GNSS-based localization methods are often insufficient for localization in terms of accuracy when adapted for autonomous driving. Positioning systems, such as GPS are subject to errors since there is no perfect method to measure location. Map-matching or filtering methods may reduce GPS errors, so that the final positioning result is sufficient enough to determine a vehicles relative position along the street (Hunter et al., 2014; Y. Li et al., 2013). However, autonomous vehicles require centimeter-level positioning precision and a detailed map for decision-making in the driving environment. In the driverless age, positional errors for more than 5 cm may be unacceptable (Goodchild, 2018). The geo-database for drivers navigation is also insufficient to support autonomous vehicles. The geo-database for drivers navigation usually consists of streets and point of interests (POIs). Street centerline (SCL) database, where the streets and road networks are represented as collections of line edges connecting nodes is a commonly used type of street database. A POI is a specific point location that may be useful or interesting for drivers. The geodatabase (SCL database and POIs) for drivers though simple but is enough for the drivers to make their decisions. Information such as turning curves, or lanes may be redundant for drivers and seldom appears in navigation databased could be critical for autonomous vehicles. Drivers can take that for granted. However, autonomous vehicles always need to know which lane

they occupy or the exact location of a parking slot instead of a parking lot POI. The geodatabase should also be compatible with the sensory system in autonomous vehicles. Autonomous vehicles are equipped with a combination of sensors such as cameras, LiDAR, GPS, IMU (inertial measurement unit), and radars to help them to perceive the driving environment. The automotive industry goes towards a high-definition map (HD map) to support multi-sensor localization and accurate streets information retrieval. Current HD maps extend the SCL database and POIs with sensory compatible data models and more detailed road and road furniture information. Productions about HD map can be found in HERE[1] , TomTom[2].

Place as a collective concept combines locations and rich platial information in geodatabase (Bennett & Agarwal, 2007; Goodchild, 2011; Winter & Freksa, 2012). It enables organizing locations and their attributes at a higher abstraction level to support geographical information retrieval (GIR) (Jones et al., 2001). Place recognition shares some common principles. Place recognition can be performed by measuring the similarity among locations in a continuous space or integrating locations of relevant semantics into a higher semantic level (Bennett & Agarwal, 2007). On the contrary, the similarity measurement or the content of a place largely depends on the specific tasks in different LBS applications (e.g., Alazzawi et al. 2012. Researchers have done extensive work on mining place names from text descriptions Vasardani et al. 2013. As the geo-tagged photos became more accessible on the Internet, mining place related information from images under the research of volunteered geographic information (VGI) began to emerge (L. Li & Goodchild, 2012). Visual features in images contain sufficient locational and semantical information. Recognition of places from images for localization and information retrieval also attract researchers from computer vision (Lowry et al., 2016).

1.1.3 Place-based Research Trigged By Big Image Data

With the rapid development of Web 2.0 and camera embedded mobile phones in the last 20 years, the number of photos on the internet is consistently increasing with rather high speed. In 2011 Flickr announced its 6 millionth photo uploading, with an annual increase of 20% since 2006[3]. "On average more than 350 million photos per day were uploaded to Facebook in the fourth quarter of 2012. Over 240 billion photos have been shared on Facebook" according to the Facebook annual report for 2012[4]. Some figures show that one trillion photos would be created in the year 2017, adding the total number of photos up to 4.7 trillion. Most of these photos are taken by low-end embedded cameras in mo-

[1]Here HD map. https://www.here.com/en/products-services/here-automotive-suite/highly-automated-driving/here-hd-live-map access date: 18.10.2018

[2]TomTom Road DNA. https://www.tomtom.com/automotive/automotive-solutions/automated-driving/hd-map-roaddna/ access date: 18.10.2018

[3]Flickr blog. https://blog.flickr.net/en/2011/08/04/6000000000 access date: 11.22.2018

[4]Facebook annual report 2012. https://investor.fb.com/financials/ access date: 11.22.2018

bile phones[5]. Figure 1.1 depicts the trends of an increasing number of digital photos over the world in recent five years and the percentage of photography devices. In addition to datasets that consist of voluntarily uploaded images, there are also purposefully collected images on the internet. For example, Google Street View has a collection of 360-degree street-level imagery from many cities. Camera systems embedded in and at cars collect most images. Geographic-coordinates and 3D depth information are measured by a positioning system, e.g., GPS and LiDAR scanners, and provided along with these images.

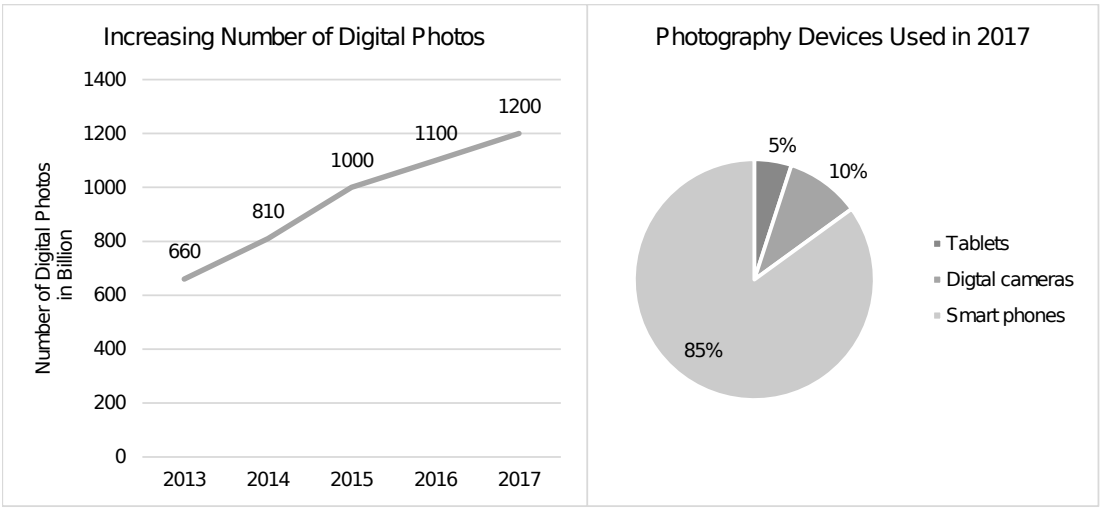


Figure 1.1: Estimated number of digital photos taken over years in the world (left), and the percentage of photography devices (right). (Data credit: Info Trends via Bitkom).

Michael F. Goodchild predicted in 2007 that the contents generated by citizens have been shaping the geographic information science (Goodchild, 2007). This trend also has a large impact on other research fields. The rich image resources on the internet have brought about new ideas in the field of computer vision. Among others, they are radically fueling deep learning, which needs huge data amount to train its deep neural networks (DNN). For example, the ImageNet[6], a famous benchmark dataset in computer vision research, contains over 1 million images, classified as 20 thousand classes, including animals, natural objects, artifact, person, and activities (Deng et al., 2009). The ImageNet Large Scale Visual Recognition challenge (ILSVRC) aims at evaluating vision algorithms in a wide range of tasks, e.g., classification, detection, localization, and so on (Russakovsky et al., 2015). Since AlexNet (Krizhevsky et al., 2012) competed in 2012 with a great score ahead, deep learning methods have been gradually taking the dominant position in this

[5] Here is How Many Digital Photos Will Be Taken in 2017. https://mylio.com/true-stories/tech-today/how-many-digital-photos-will-be-taken-2017-repost access date: 11.22.2018

[6] ImageNet. http://www.image-net.org/ access date: 11.22.2018

challenge. In a recently published dataset, Google released more than 2 million images depicting 30 thousand unique natural or human-made landmarks from across the world. The dataset aims at advancing instant-level landmark recognition (Noh et al., 2016). Benefiting from the rapidly advancing machine learning and computer vision, applying data-driven research methods to large scale image data brings a new view to place related concepts. Google Street View is a widely used data source in researches, such as place recognition (Arandjelovic et al., 2016). Street-level images are also used to predict different types of human perceptions of a place when combined with human ratings (L. Liu et al., 2016; F. Zhang et al., 2018). Visual properties that learned from street level images by using deep learning algorithms are used to identify characters of different cities (B. Zhou et al., 2014). Autonomous driving is one of the most benefiting research fields that require both huge image datasets for various visual perception tasks, and powerful deep learning tasks to achieve high correctness and effectiveness. For autonomous driving, cameras are widely equipped sensors for environment perception (Ziegler et al., 2014). As the autonomous vehicle driving, its camera system captures a huge amount of images. Perception of the environment includes vision tasks of motion estimation, object segmentation, vehicle and pedestrian recognition, and traffic sign recognition. The success of deep learning in computer vision encourages autonomous driving collects and utilizes images as the training dataset. These images are recorded along a certain road spatially and in time dimension consecutively. Widely used datasets, for example, are the KITTI vision benchmark suite[7] (Geiger et al. 2012) and Cityscapes[8] (Cordts et al. 2016).

1.1.4 Deep Learning, a Bridge between Images and Places

Deep learning is a family of computational models composed of non-linear processing units arranged in multi-layer structures to learn representations from data for specific tasks. Deep learning methods have achieved state-of-the-art results in many domains including image recognition, speech recognition, and translation (LeCun et al., 2015). The embryo of deep learning was developed and inspired by the study of the biological nervous system. Pioneering achievements include McCulloch-Pitts Neurons, which introduces neurons and neural networks as computational model(McCulloch & Pitts, 1943); Hebbian learning rule, the first learning rule for self-organized systems (Hebb, 1949); and Rosenblatt perceptron, the first supervised learning model (Rosenblatt, 1958). Then it evolved into artificial neural networks in the name of connectionism in the 1980s and 1990s. The work of Werbos (1982) and Rumelhart et al. (1986) enables the training of a neural network through back-propagating errors. Being rejuvenated as a powerful method in machine learning from a huge amount of raw input data, deep learning is often developed for making data-driven predictions and decisions other than following pre-defined static instructions. With the networks going deeper and large training data being available, learning features from a massive amount of data for a specific task became more and more popular and outper-

[7]KITTI. http://www.cvlibs.net/datasets/kitti/ access date: 11.22.2018
[8]Cityscapes. https://www.cityscapes-dataset.com/ access date: 04.25.2018

formed the traditional ones in many tasks.

Image understanding is a strong application domain where deep learning achieves amazing results. The basic task of image understanding is, for short, is to let a computer know "what objects exist where" in an image. We summarize some relevant sub-tasks here,

- *Camera pose estimation* and *3D reconstruction* aim at recovering camera pose and 3D structure of a scene from image measurements (R. Hartley & Zisserman, 2004). Structure from motion (SfM) and multi-view geometry are two basic techniques to solve the problem. It also has an important application in robotic navigation and autonomous driving.

- *Object recognition and detection* are performed at different granularity. At a rough level, it is often regarded as a classification problem, i.e., classifying a dominant object in an image into a known class, such as dogs or cats. The location of the object is required and usually represented by a bounding box. In object detection, the number of objects is not fixed, and thus several labeled bounding boxes should be predicted to indicate "what exists where" in an image.

- *Semantic segmentation*, in contrast with object recognition and detection, aims at pixel level image understanding. It requires the computer to assign each pixel with a semantic class. In practice, the specific application decides which type of semantic information should be recorded. It can be a discrete class type such as human, animal, or vehicle, to be distinguished as foreground objects from the background, or a continuous value such as depth information, height, or optical flow for each pixel. Examples can be found in the research works from (Dosovitskiy et al., 2015; Eigen et al., 2014; Horn & Schunck, 1981; Long et al., 2015; Mou & Zhu, 2018).

- *Content-based image retrieval* is a technique of indexing and searching images from a large scale image database based on the visual content of images other than any of their metadata such as keywords, tags, or other text information. Given a query image an image retrieval system is supposed to return all images with the similar content (Fei-Fei & Perona, 2005; Sivic & Zisserman, 2003).

- *Image synthesis* is a technique of creating new images from various image descriptions. The description can be text, noise without explicit semantic meanings for each element, another image, an image in a coarse resolution, or image with noise (Gatys et al., 2016; Jain & Seung, 2009; H. Zhang et al., 2017).

Before deep learning methods achieve amazing results in image understanding tasks, hand-crafted feature detectors are widely used to characterize images. These feature detectors include *SIFT* (scale-invariant feature transform) (Lowe, 1999, 2004), *SURF* (speed up robust feature) (Bay et al., 2008, 2006), *FAST* (features from accelerated segment

test) (Rosten & Drummond, 2006), *BRIEF*(Binary Robust Independent Elementary Features)(Calonder et al., 2010), *ORB* (Oriented Fast and rotated BRIEF) (Rublee et al., 2011), etc.. Numerous images can thus be represented with a manageable number of the detected features. Different from using such conventional feature detectors, convolutional neural networks (CNN) learn feature detectors in an optimization process in a specific task and use non-linearity functions to boost up their representation ability.

1.2 Research Questions and Objectives

With the notion of point set theory, we can formalize the relations between a place and its observations. Suppose a distinctive observation of a place is associated with a specific geographic location and orientation, then a place can be depicted as a set of all observations. Particularly the geometric boundary of a place can be then inferred from the geographic locations of the observations. Based on the above discussion, we have our hypothesis for learning a place representation from observations,

- A place has a latent representation, which can be learned from observation data. The geographic location of observation data is closely related to the formation of place, however, the place can be hardly defined purely by specific geographic locations.
- Each observation can be associated with a latent code. The latent code is inferred from the latent representation of its corresponding place. Observations the same place are close in the latent space; observations from different places are far away (as far as they are identified as from two distinct places) from each other in the latent space.
- Geographically closed observations are probably taken from the same place.

Cognitively, salient tangible objects can provide strong visual clues to distinguish between different places. We assume that both spatial and visual properties contribute to the formation of "place". Imagery observations capture the visual property of a place from different views, and their camera poses reflects how spatial property interacts with the visual property. Since the cameras systems are extensively embedded in various machines (e.g., mobile devices, autonomous machines), investigate the spatial and visual property of a place is important to support intelligent behaviors, such as querying geo-information, localizing and navigating autonomous vehicles and robots. Conventional techniques, such as structure from motion (SfM) and multi(two)-view geometry, recover the geometric relationship between pixels in image and points in 3D scenes. They produce point-based representations such as point clouds about the 3D structure of the environment. Handcrafted features are often used to characterize the visual property of these points. With the booming of deep neural networks, extracting visual features by learning becomes more and more popular. The visual features learned by deep networks are thought to be able to capture meaningful semantics at a high abstraction level.

The main research question is formed as "how to approach a place definition that connects visual and spatial/semantic property of the place from a large number of imagery observations?" In order to answer the main research question, there are three sub-questions to be answered,

1. How to model the latent place representation for connecting visual and spatial/semantic property from a data-driven perspective?
2. How to learn the latent place representation model from large imagery observations with the help of deep learning methods?
3. To what extent the learned visual representation can support specific spatial tasks such as place recognition?

The objectives of this dissertation are three folds:

1. Investigate the role of geo-location and visual features in modeling and forming the concept of a place.
2. Find possible data-driven methods and models that are adequate for learning place representation.
3. Identify the advantages and limitations of the data-driven based approach to understand a place.

1.3 Thesis Structure

This work aims at expanding the understanding of "place" from a data-driven perspective with the help of advanced machine learning methods. In Chapter 2, we briefly introduce the technical foundations of this work with the efforts on the mechanism behind deep learning. We also show the connection between images and their camera pose. Camera pose reflects the location and the orientation of an observer and thus bringing together the information of a place and the information of the observer in the image. In Chapter 3, we first review recent advances in deep learning specifically on deep neural networks including deep convolutional networks (deep CNN) and deep generative models. Deep convolutional networks provide powerful tools to extract visual features for various vision tasks as well as for our research. Generative models can jointly learn data distribution and the distribution of a latent variable. In our research, it serves as a bridge connecting the image data and the abstract concept of place. In the rest part of Chapter 3, we review the two most relevant tasks of our work, place recognition and pose estimation. Most of these works aim at enabling a machine to retrieve locational and positional information from visual input. We roughly divided these works into two parts, researches on geometric methods aim at rebuilding the consistency of geometric relationships from two or more consecutive images to get camera pose and the 3D structure of a scene; while the learning

methods are stronger in image understanding at the semantic level. We make our efforts to reveal the influence of the rapidly developing deep learning as a bridge connecting the two types of methods in place recognition and pose estimation.

In Chapter 4, we describe the methodology in a theoretical framework of place definition and then derive our specific approach from a data-driven perspective for a collective place definition. In this chapter, we propose a probabilistic place description based on the latent variable model and introduce two learning techniques, i.e., variational autoencoders (VAE) and comparative learning to implement the proposed approaches. Then we introduce the computational models that we derived from VAE and comparative learning methods to learn the latent place representation. The methodological road map is shown in Figure 1.2. To summarize, we first derive the specific methods to approach a collective place definition from the general framework by Cresswell (Cresswell, 2014). These methods are implemented via deep learning models. Three alternative functional evaluations are carried out to check whether the learned place achieves the general and collective place definition. In Chapter 5 we show several experiments regarding the functional evaluations for the proposed methods. In the concluding chapter, i.e., Chapter 6, we summarize our work and discuss whether such learning based place definition approach can be used as a bridge connecting human cognition and machine cognition in terms of spatial knowledge. Some future directions are given as well.

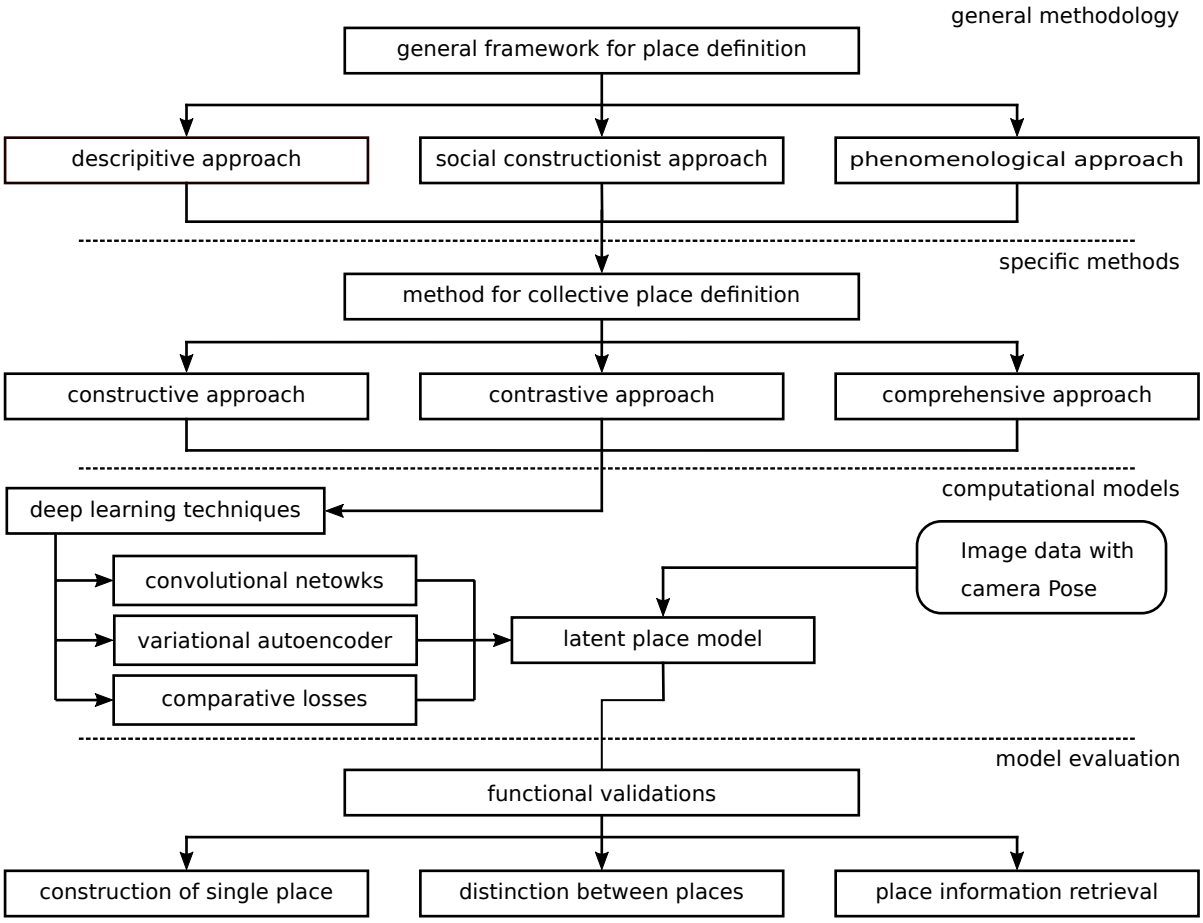


Figure 1.2: The road map of the research methodology used in this work.

Chapter 2

Fundamentals of Data-Driven Place Learning

Deep learning as an advanced data-driven method plays a key role in this research. This chapter builds a context for understanding the methodology and experiment by introducing the most relevant terms and concepts in machine learning, neural networks, and camera imaging. In the first part of this chapter, we introduce the basics of deep learning under the umbrella of machine learning. In the second part, we put our effort on the fundamental operators in deep convolutional neural networks and their functions in visual feature extraction and image reconstruction. Back-propagation, the backbone algorithm for training deep networks is briefly introduced as well. Detailed information about the three topics can be found in corresponding textbooks (we referred some in this chapter). In the last part, we show the relationship between camera poses and observed images, because camera pose (including geographic location and observing orientation) is important auxiliary information that connects the spatial and visual properties of a place. Though most of the content in this chapter can be found in textbooks, we put the most relevant and fundamental knowledge in this chapter to make this dissertation self-contained and barrier-free for readers who are not familiar with these things.

2.1 Machine Learning Basics

This section presents a brief introduction about deep learning basics. Important terms, concepts, and techniques in machine learning are introduced to build up the context for the rest of this thesis. The topics mentioned in this chapter are discussed in detail in many textbooks on machine learning. The recent published "Deep Learning" book from Goodfellow et al. (2016) is recommended as comprehensive reading material in the deep learning field.

2.1.1 Learning as an Optimization Problem

Deep learning is a specific machine learning method, sharing a large portion of common basics with other machine learning methods. In terms of learning in machines, we find the definition by Mitchell (1997) as:

"a computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**."

Classification and regression are two typical tasks in machine learning. Classification is the task of assigning each observation into one of k categories. Regression requires the learning algorithm to produce a numerical value over the input. There are also other machine learning tasks, such as synthesis and sampling, denoising, and density estimation.

Qualitative measurement P is used to evaluate the performance of learning for the specific task T. Take the task of classification as an example, the correctness of classification is a widely used evaluation metric. Counting the number of correctly classified data samples could be a measurement, or we can calculate the portion of correctly classified ones as the accuracy of the algorithm. If the desired output is available in training dataset, we can calculate the Mean Squared Error (MSE) as the squared value of the difference between the output value from the algorithm and the desired value (Equation 2.1):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \|y_i - \hat{y}_i\|^2, \tag{2.1}$$

where n is the number of data samples, y_i is the desired output for the i^{th} input data sample, and $\hat{y}_i$ is the prediction made by the algorithm.

Machine learning algorithms generally get experience **E** in two paradigms, i.e., supervised and unsupervised, where a major difference is whether the desired output is provided during training or not.

In **supervised learning**, each training sample x is given with a label y as its desired output. The supervised learning task is to find out a function that approximates the true conditional probability $p(y|\boldsymbol{x})$, given $\boldsymbol{x}$. A deep model can be seen as a parametric function family $\mathcal{F}$ characterized by parameters $\boldsymbol{\theta}$. Each parameter corresponds to a function that maps input $\boldsymbol{x}$ to some value $\hat{y}$ under the parameter set, i.e., Equation 2.2. The learning algorithm aims to find an optimal parameter $\boldsymbol{\theta}^\star$ such that the resulted function produces a probability $p(y|\boldsymbol{x};\boldsymbol{\theta}^\star)$ that is close enough to the true conditional probability $p(y|\boldsymbol{x})$.

$$\hat{y} = f(\boldsymbol{x};\boldsymbol{\theta}) = p(y|\boldsymbol{x};\boldsymbol{\theta}). \tag{2.2}$$

A loss function $\mathcal{L}$ is used to measure the difference between estimated probability and the true probability that we want to minimize. Take classification task as an example, usually softmax activation (see Equation 2.3) is used to get the probabilities on each class and the cross-entropy loss 2.4 is used to measure the performance. The training procedure is needed to indicate how to adjust the trainable parameters to minimize the loss function. Suppose there are m data samples associated with labels drawn from an **i.i.d.** (independent and identically distributed) but unknown data-generating distribution $p_{\text{data}}(\boldsymbol{x}|y)$, $\{\langle \boldsymbol{x}^{(1)}, y^{(1)}\rangle, \cdots, \langle \boldsymbol{x}^{(i)}, y^{(i)}\rangle, \cdots, \langle \boldsymbol{x}^{(m)}, y^{(m)}\rangle\}$. Let $p_{\text{model}}(\hat{y}|\boldsymbol{x};\boldsymbol{\theta})$ denotes the $\boldsymbol{\theta}$ indexed parametric family of probability distributions, which produces a value for each sample $\boldsymbol{x}^{(i)}$ as an estimation of the true probability $p_{\text{data}}(y|\boldsymbol{x})$. We assume the existing dataset is the most possible existence of $\boldsymbol{x}$, and thus a surrogate to the true distribution. We can thus use a maximum likelihood estimator for $\boldsymbol{\theta}$, which is defined as Equation 2.5 or using an equivalent logarithmic form as Equation 2.6.

$$p(y=j|\boldsymbol{x}) = \frac{e^{\boldsymbol{z}_j}}{\sum_{k=1}^{K} e^{z_k}}, \qquad \text{for} j = 1, \cdots, K. \tag{2.3}$$

$$\mathcal{L}_{\text{cross\_entropy}} = -\sum_{j=1}^{K} p(y=j)\log p(\hat{y}=j). \tag{2.4}$$

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}(y^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}), \tag{2.5}$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}(y^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{2.6}$$

Alternatively, we can also use maximum a posteriori (MAP) to get $\boldsymbol{\theta}$, if we have a prior distribution (Equation 2.7 and 2.8).

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p(\boldsymbol{\theta}|\boldsymbol{x}^{(i)}, y^{(i)}), \tag{2.7}$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p(\boldsymbol{x}^{(i)}, y^{(i)}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}). \tag{2.8}$$

Unsupervised learning includes a wide spectrum of scenarios including probability density estimation, clustering, and reinforcement learning (see Sutton and Barto 1998). Unsupervised learning includes tasks such as density estimation, denoise, and data synthesis. Here we only focus on the probability density estimation, a topic that is closely related to our work. To learn the probability distribution $p(\boldsymbol{x})$ over a dataset $\mathbf{X}$, the algorithms must discover the internal structure of the dataset without explicit labels. Unsupervised learning can be applied in a scene that we have observed some samples of a random vector $\boldsymbol{x}$ and attempt to model the probability distribution $p(\boldsymbol{x})$ or some interesting properties of

that distribution. Using the chain rule of probability states, we can decompose the joint distribution over the whole dataset as

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(\boldsymbol{x}_1, \cdots .\boldsymbol{x}_{i-1}). \tag{2.9}$$

This decomposition enables to turn the unsupervised problem into n supervised learning problems by splitting it.

2.1.2 Gradient Descent Method

The gradient-based method is a widely used optimization technique. The basic ideas of Gradient descent methods came from the observation that the minimum (local minimum) of a differentiable loss function can be achieved by iteratively moving along the negative direction of its gradient. Suppose we want to minimize a function $f(\boldsymbol{x})$. Expanding the function in Tylor series, we will have a first order approximation at the neighborhood of $\boldsymbol{x}$, i.e.,

$$f(\boldsymbol{x} + \epsilon) \approx f(\boldsymbol{x}) + \epsilon f'(\boldsymbol{x}). \tag{2.10}$$

The sign of $f'(\boldsymbol{x})$ indicates to which direction we move for a small step $\epsilon f'(\boldsymbol{x})$ will decrease f. If the function is convex, it is guaranteed to find a global minimum point. It is a necessary condition that the first order derivation is zero at the optimal point (i.e., $f'(\boldsymbol{x}^\star) = 0$). However, zero derivation point can also be saddle point that is neither (local) maxima nor (local) minima. We show different point types in Figure 2.1. In deep learning cases where the objective function is non-convex, there are lots of local minimum (maximum) points and saddle points that make the optimization difficult. In our applications, it is sufficient to find a local minimum point that makes the object function "low enough".

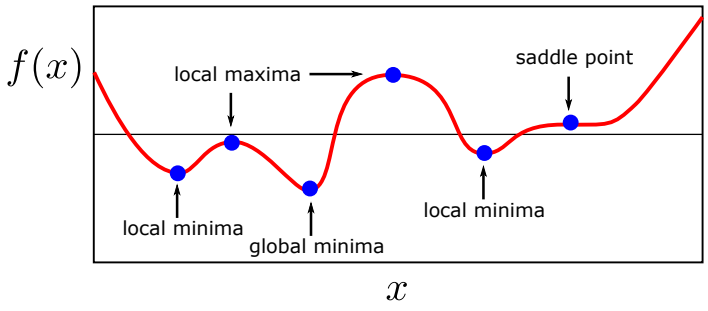


Figure 2.1: An example of different point types on the plot of function $f(x)$, including local minima, local maxima, global minima, and saddle point.

In deep learning, there is always a loss function (also called as objective function, cost function, or error function) $f(\boldsymbol{x};\boldsymbol{\theta})$ to be optimized. This function must be continuously differentiable for some parameter $\boldsymbol{\theta}$. A learning algorithm aims to find an optimal $\boldsymbol{\theta}^*$ such that for every input $\boldsymbol{x}$, the objective function produces an minimum (or maximum) value, i.e., $f(\boldsymbol{x};\boldsymbol{\theta}^*) \leqslant f(\boldsymbol{x};\boldsymbol{\theta})$ for the minimum optimization case. The gradient $\nabla f = \left[\frac{\partial}{\partial\theta_1}, \frac{\partial}{\partial\theta_2}, \cdots, \frac{\partial}{\partial\theta_M}\right] f$ of the function indicates the direction in which f decreases the fastest. Thus gradient descent method is also called steepest descent.

Gradient-based methods start from an initial point in the parameter space and search its neighborhood to find another point with the decreased objective function value. The methods usually choose a new point $\boldsymbol{\theta}+\Delta\boldsymbol{\theta}$ along the negative gradient direction of point $\boldsymbol{\theta}$ as shown in Equation 2.11, with a properly small step-size ϵ. If such operation is iteratively performed, i.e., for the n^{th} iteration $f(\boldsymbol{\theta}_{n+1}) < f(\boldsymbol{\theta}_n)$, it can be expected to find a point where the gradient is zero. If the function f is convex, gradient descent is guaranteed to find the global optimal value. However in cases when f is non-convex, the point where the gradient is zero can be a local minimum or a saddle point. In cases where a method cannot find a zero gradient point, it fails the learning as being divergent.

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \epsilon_{n+1}\nabla f(\boldsymbol{\theta}_n). \tag{2.11}$$

2.1.3 Stochastic Gradient Descent (SGD)

The loss function used in deep learning is often formulated as the average of the cost per training sample. The average loss can be seen as the expectation when $\boldsymbol{x}$ and y obey the modelled distribution p_{model} as well (see Equation 2.12).

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y\sim p_{\text{model}}} L(\boldsymbol{x}, y, \boldsymbol{\theta}). \tag{2.12}$$

where L is the cost for each sample. The cost function $\mathcal{L}$ can be, for example cross-entropy loss (recall Equation 2.4), and the corresponding gradient is given as the following (Equation 2.13),

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y\sim p_{\text{model}}} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}, y, \boldsymbol{\theta}), \tag{2.13}$$

In stochastic gradient descent, the expectation is approximated by a single sample or a small portion of the whole samples, i.e., **mimi-batch** of samples. Thus the expectation of gradient is approximated as Equation 2.14:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m'}\sum_{i=1}^{m'} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{\theta}) = \hat{\mathbb{E}}_{\boldsymbol{x},y\sim p_{\text{model}}} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}), \tag{2.14}$$

where m' is usually much smaller than the total number of samples m. SGD is a better mechanism to catch the information in the dataset than the non-stochastic version, especially when there is much redundant information. When calculating with a large number of samples, SGD can keep a balance between precision and computation cost. More details we refer to the 4th part of Bottou's work (Bottou et al., 2016).

Learning rate ϵ, a hyperparameter which is given priorly to determine how "fast" an algorithm learns. If it is set too large or too small, the learning algorithm may miss the optimal point or converge to slow. Especially in the late stage of learning, we often want a smaller learning rate to ensure the algorithm does not miss the minimum point. Many advanced gradient descent algorithms try to avoid such risk such as Momentum, AdaGrad, RMSprop, Adaptive moment estimation (Adam). More introductory information about gradient descent optimization algorithms we refer to Ruder's work[1] (Ruder, 2016).

2.2 Introduction to Neural Networks

There are two major types of artificial neural networks, the feed-forward network in which information flows from the input neurons directly to the output, and the recurrent neural network (RNN), which consists of a feed-forward network taking its output as part of its input. In this section, we focus on a specific type of feed-forward network, convolutional networks, as it is the major part of the learning model in our work.

2.2.1 Elements of Artificial Neural Network (ANN)

An artificial neural network is composed of many simple computing units as an analogue to a biological neuron. The computing unit is often referred as neuron in the context of ANN if it is not misleading with the biological neuron. In Figure 2.2, we present a schematic drawing of a neuron used in ANNs. The neuron receives information from all its connected predecessors, and spreads its output to all connected successors. The neuron presented in Figure 2.2, marked as k^{th}, receives m inputs, $x_1, x_2, \cdots, x_m$. Each input is multiplied with a corresponding weight w_i. The weighted sum of these inputs (including a bias b_k) v_k is then passed through an activation function σ. The computation process can be described by Equation 2.15,

$$y_k = \sigma(\sum_{i=1}^{m} w_{ki} \cdot x_i) + b_k. \tag{2.15}$$

The activation function adds non-linearity to a neural network. Figure 2.3 presents several commonly used activation function, such as Tanh $(1 - e^{-2x})/(1 + e^{-2x})$, the sigmoid function $1/(1 + e^{-x})$, and the rectified linear unit (ReLU) $max(0, x)$. The output of a Tanh

[1] http://ruder.io/optimizing-gradient-descent/index.html access date: 05.12.2018

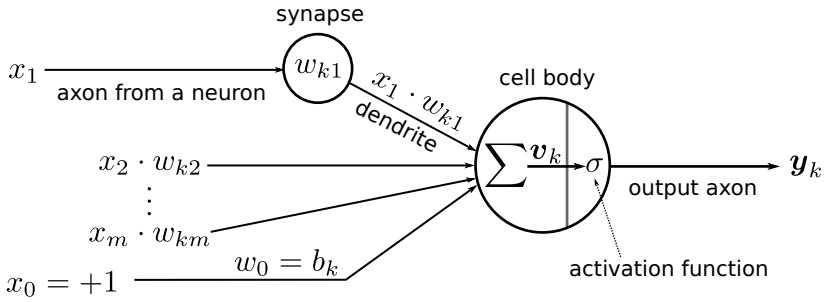


Figure 2.2: The schematic drawing of an artificial neuron, which takes input from the output of its predecessors and produces a scalar output.

ranges from -1 to 1, which can be used in image generation to produce normalized pixel values. Sigmoid that squeezes input into [0, 1] can be used to produce a probability. ReLU is also widely used in the hidden layers of deep neural networks.

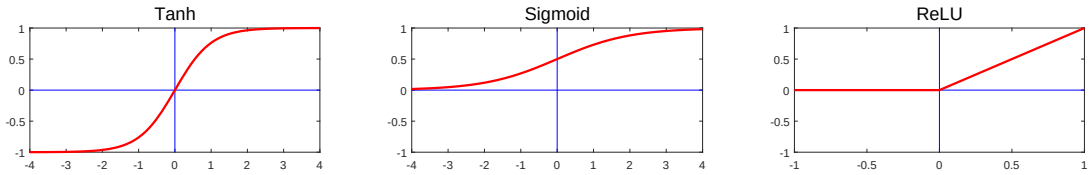


Figure 2.3: Graphic explanation for activations functions, i).Tanh, ii).Sigmoid, iii). ReLU.

Generally, an artificial neural network consists of an input layer, one or more hidden layer(s), and an output layer. Each layer consists of a set of neurons. Neurons in different layers are connected, while neurons in the same layer are not. The phase "fully connected" indicates that the neurons in one layer have connections to every neuron in the successor layer. With vector and matrix notations, the computation in a neural network can be represented by repeating matrix multiplication and element-wise non-linearity. In Figure 2.4, we represent three layers L, $L+1$, and $L+2$ in green, purple, and light red respectively, where $L+1$ is fully connected with both L and $L+2$. Let $\boldsymbol{u}^{(\ell)}$ and $\boldsymbol{u}^{(\ell+1)}$ denotes the output of layer L and $L+1$, the output of layer $L+2$ is written as Equation 2.16, 2.17,

$$\boldsymbol{u}^{(\ell+2)} = \sigma(\boldsymbol{w}^{(\ell+2)} \cdot \boldsymbol{u}^{(\ell+1)}) + \boldsymbol{b}^{(\ell+2)}, \tag{2.16}$$

$$= \sigma(\boldsymbol{w}^{(\ell+2)} \cdot \sigma(\boldsymbol{w}^{(\ell+1)} \cdot \boldsymbol{u}^{(\ell)} + \boldsymbol{b}^{(\ell+1)}) + \boldsymbol{b}^{(\ell+2)}), \tag{2.17}$$

where σ is a non-linear activation function, $\boldsymbol{w}$ and $\boldsymbol{b}$ are weights and bias in each layer.

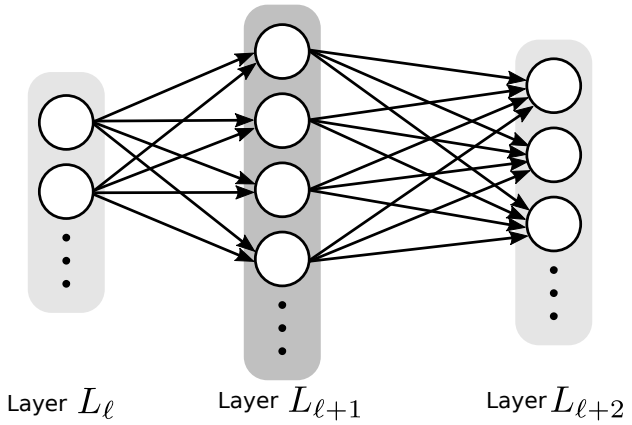


Figure 2.4: An example of three layers L, L+1, and L+2 in a neural network where layer L+1 is fully connected with layer L and L+2.

2.2.2 Convolutional Neural Networks

The convolutional neural network, also mentioned as CNN or ConvNets, is a neural network model that is particularly designed for uniformly sampled data, such as voice and imagery data. A typical convolutional layer consists of three components in a cascaded arrangement. The three components are convolution, activation, and pooling. Sometimes the three components are seen as three simple layers, i.e., convolutional layer, activation layer, and pooling layer. In this section, we introduce the three layers, together with up-sampling operations (for image reconstruction) in the context of handling image data.

Convolution and convolutional layer

Mathematically, convolution is an operation on two functions to produce a third function. For images we specially need discrete convolution in two dimensions. Let $I(u, v)$ denotes an image function which indexes pixel values with u and v, its convolution with a discrete filter function ϕ is written as:

$$f(u,v) = I(u,v) * \phi(u,v) \doteq \sum_m \sum_n I(m,n)\phi(u-m, v-n), \tag{2.18}$$

$$= I(u,v) * \phi(u,v) \doteq \sum_m \sum_n I(u-m, v-n)\phi(m,n). \tag{2.19}$$

Convolution on images can be understood by using sliding window method. The filter is referred as convolution **kernel**, and set kernel size to control the size of its neighboring area. **Stride** controls the step length of the sliding window. With stride greater than 1, the output size is reduced. **Padding** is used to indicate how to handle the edge situation. If only valid image area is used in convolution, the output size is reduced. If the image is expanded with designated values, the output can keep the same size as input. Figure 2.5 gives an example of image convolution. The upper left 3×3 matrix represents a

convolutional kernel with size 3. The kernel slides along horizontal and vertical directions on an image with stride equals to 1. With the case that valid image area is used in convolution, the output size is reduced by *kernel size -1*. At each location, the filter is element-wisely multiplied and summed with the corresponding image region to produce a value for the location.

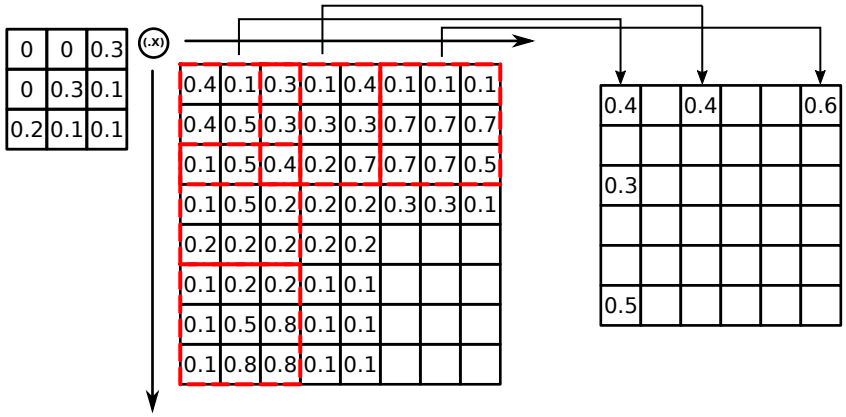


Figure 2.5: An explanation of image convolution by showing sliding kernel along horizontal and vertical directions. The symbol (.x) stands for the discrete convolution operation, i.e., an element-wise multiplication followed by a summation.

There is another view of convolution, where the kernels are expanded into a sparse matrix, input and output are flattened. Figure 2.6 gives a simple example with kernel size equals to 3, 4×4 input and 2×2 output. The kennel is expanded as a 4×16 matrix, the input is flattened as a 1×16 matrix, and the output is expected to be 1×4 by multiplying the expanded kernel and flattened input.

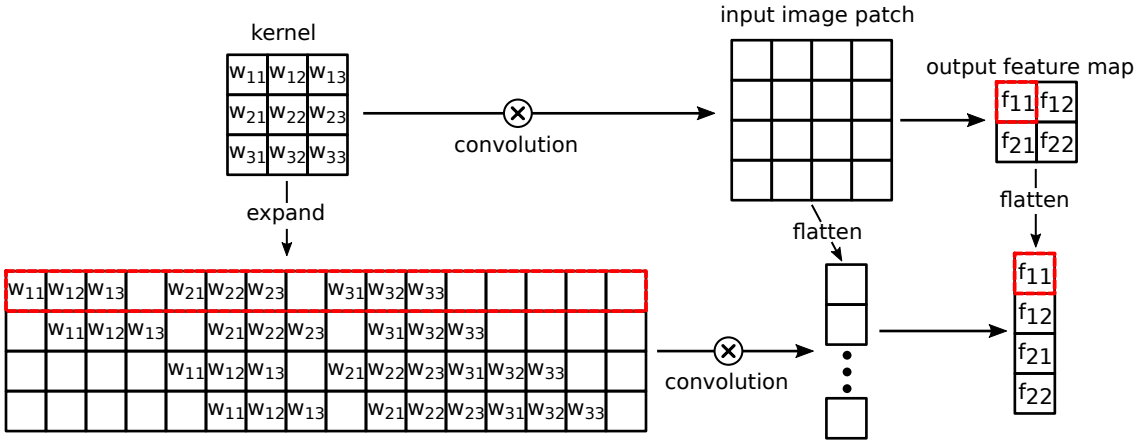


Figure 2.6: An example of image convolution with expanded kernel and flattened input.

Convolutional layer is the most important component in CNNs. The filters in convolutional layer are referred as **kernel** functions. The output of a convolutional layer is often called as **feature map**. Though an efficient implementation of convolution can be very different, we can intuitively imagine the kernel slides over the whole input image to produce a feature map. A convolutional layer takes a $B \times H \times W \times C$ tensor as input, where H, W, and C, denote the height, width, channel (or depth) of the input (either image or feature map), and B stands for the number of images in a batch. For a single image input, the batch size dimension can be omitted. The layer uses a $k \times k \times C \times D$ tensor as its kernel (k represents the kernel size, C, and D are the input depth and output depth respectively). We have two views to understand the convolution operation in terms of decomposing the input into a stack of feature maps or spatially bundled fibers. When sliding the kernel on the input feature map, in each step it operates on a slice of input with shape $k \times k \times C$. In the first decomposition, the sliced tensor is C vectors of $k \times k$ elements, and the kernel is C pieces of two-dimensional tensors with shape $[D, k \times k]$. The affine transformation is then applied to each vector with the corresponding tensor resulting in a vector with D elements. The final result comes by summing up all C vectors element-wisely. In the fiber decomposition, the sliced tensor is $k \times k$ vectors of C elements, and the kernel is $k \times k$ pieces of two-dimensional tensors with shape $[D, C]$. Applying affine transformation and then element-wise adding, results in the same vector. Figure 2.7 illustrates convolution on the two decompositions. To emphasize the difference, each piece of feature maps is kept as a $k \times k$ matrix other than a flattened vector. The kernel is not flattened as well.

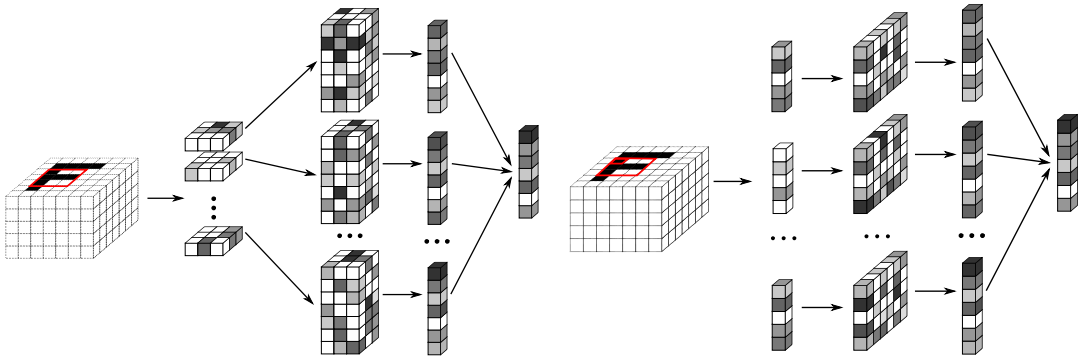


Figure 2.7: Two decompositions on performing convolution operation. The input tensor has shape [7, 7, 5], with a [3 × 3, 5] slice for convolution. **Left**: The slice is decomposed into 5 feature maps of the shape [3, 3] and the kernel is reshaped as 5 matrices of shape [3 × 3, 7]. The result is a vector of 7 elements **Right**: The slice is decomposed into a bundle of 3 × 3 fibers of 5 elements, and the kernel is reshaped as 3 × 3 matrices of shape [5, 7]. The convolution results the same vector of 7 elements.

Transpose convolution (TransConv) layer

When we use neural networks to generate images, we need to up-sample from low resolution to high resolution. Conventional interpolation methods, such as nearest neighbor interpolation, bi-linear interpolation, bi-cubic interpolation are training-free up-sampling methods. Transpose convolution (sometimes is inappropriately mentioned as deconvolution) for up-sampling has trainable parameters. For example, the authors of DCGAN use transpose convolution to generate images (Radford et al., 2015). There are also learning-based up-sampling methods, such as PixelShuffle (Shi et al., 2016), but transpose convolution is a simple and effective one.

Figure 2.8 gives a simple example of transpose convolution. The 3×3 convolutional kernel is expanded as a sparse matrix of 16×4 dimension which can be seen as the transpose of the kernel in normal convolution. The transpose convolution is similar to normal convolution as they produce the output with the same resolution. However, when we set the stride greater than 1, for example, 2, the output resolution is scaled up by 2. In this case we up-sample a $m \times m$ input to a $2m \times 2m$. In a transpose convolution layer, the neurons have the same connectivity as in the normal convolutional layer but with the backward direction.

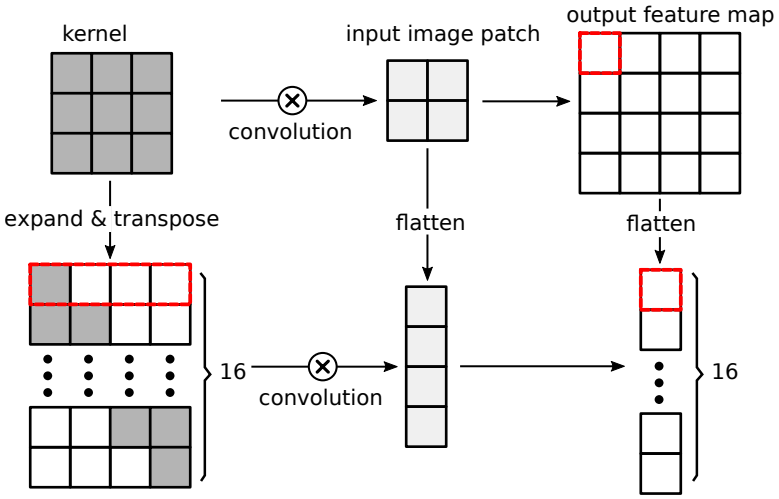


Figure 2.8: Transpose convolution with a 3×3 kernel,2×2 input and a 4×4 output. Compare to Figure 2.6, the expanded kernel is transposed.

Pooling layer

Pooling layer is used to abstract information in a neighboring area to control overfitting and further reduce the number of parameters. The pooling function affects independently on every depth slice of the input and resizes it spatially. Commonly used pooling types include maximum pooling and average pooling. Figure 2.9 shows an example of a most common pooling with window size 2×2 and a stride of 2. The pooling layer with such settings down-samples every depth slice in the input by *2* along both width and height, discarding 3/4 of the values in inputs. The pooling operation enables a CNN invariant to some minor translation. However, we can replace pooling by a convolutional layer with a larger stride to have certain benefits as mentioned by Springenberg et al. (2014). A larger stride of convolution kernels also results in reduced size of the output.

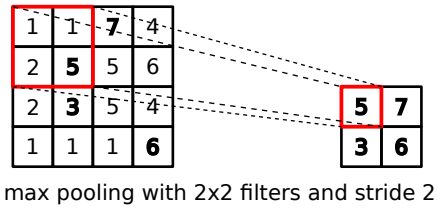


Figure 2.9: An example of max pooling with 2×2 filter and stride 2.

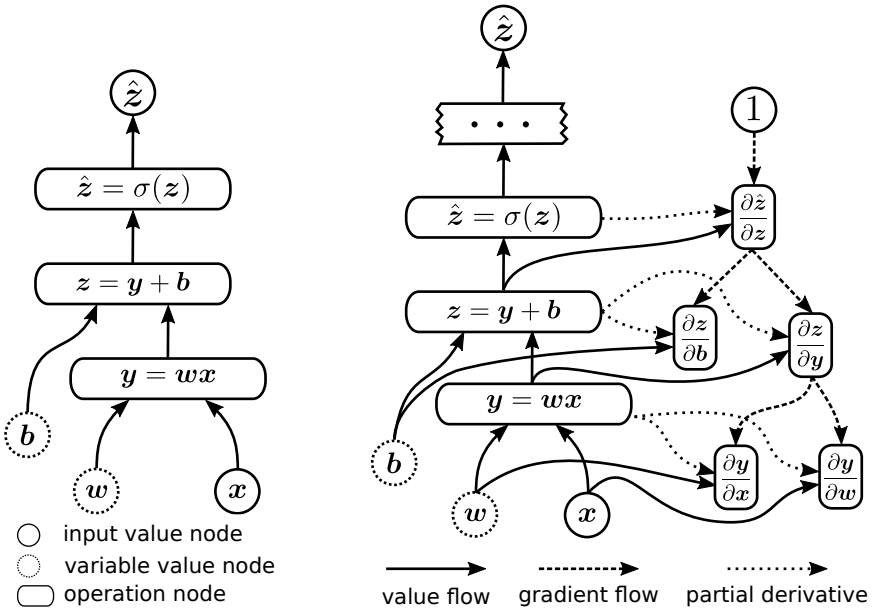
2.2.3 Back-propagation

Back-propagation is a method for computing gradients in neural networks. It is especially important in the optimization of a neural network. During a gradient descent optimization, the "error information" given by a loss function is back-propagated through layers, with the gradient of each trainable weight. In this way, the algorithm adjusts the weights in each layer according to their gradients. In this section, we first introduce the computational graph as a formal language to describe computation processes. Then we illustrate how to apply the chain rule of calculus to implement back-propagation.

Computational graph

The computational graph is a formal language to precisely describe computations. Regardless of variations of computational graphs, we present one possible implementation here. The graph consists of two types of nodes. The nodes that hold values, including scalars, vectors, tensor, or other types, are called **state nodes**; the other type of nodes are called **operation nodes**. Operation nodes are simple functions that take one or more values from state nodes as input. Thus a directed acyclic graph is built by adding directed edges between nodes to indicate how values floating from one node as output to another node as input.

Take the following function as a simple example. Let $\boldsymbol{z} = f(\boldsymbol{x}) = \boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b}$, where $*$ represent matrix multiplication, $+$ is element-wise adding; $\boldsymbol{x}$ and $\boldsymbol{w}$ are compatible tensors; $\boldsymbol{b}$ and $\boldsymbol{z}$ are output values. Let $\boldsymbol{y} = \boldsymbol{w} * \boldsymbol{x}$, this function can be seen as a composition of a tensor multiplication and a vector adding. Then we construct a computational graph as shown in Figure 2.10(a), including two operation nodes, matrix multiplication and adding (represented as round corner rectangles), and four state nodes (represented as circles) holding the four values involved.



(a) (b)

Figure 2.10: The computational graph (a) of $\boldsymbol{z} = f(\boldsymbol{x}) = \boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b}$ representing a layer of a neural network, and its back-propagation schema (b). (a) a computation graph example; (b) the corresponding back-propagation schema.

Chain rule of calculus

The chain rule of calculus is used to compute the derivatives of functions that are composed by two or more functions, whose derivatives are known. That is, if f and g are two functions, then the chain rule expresses the derivative of their composition $f \circ g$ in terms of the derivatives of f' and g' is $(f \circ g)' = (f' \circ g) \cdot g'$. If x is a real number, and f, g each maps a real number to a real number, let $y = g(x)$ and $z = f(y)$, the chain rule is represented as the following:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}z}{\mathrm{d}y}\frac{\mathrm{d}y}{\mathrm{d}x}. \tag{2.20}$$

For vectors $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^n$, if there are two functions $g : \boldsymbol{x} \to \boldsymbol{y}$ and $f : \boldsymbol{y} \to z$ ($z \in \mathbb{R}$), i.e., $\boldsymbol{y} = g(\boldsymbol{x})$ and $z = f(\boldsymbol{y})$, then the chain rule is represented as:

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i}, \tag{2.21}$$

or written equivalently using vector notation:

$$\nabla_{\boldsymbol{x}} z = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^T \nabla_{\boldsymbol{y}} z, \tag{2.22}$$

where $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ is $n \times m$ Jacobian matrix of g. The gradient of a variable $\boldsymbol{x}$ can be obtained by multiplying the Jacobian matrix $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ by the gradient $\nabla_{\boldsymbol{y}} z$. For functions that take tensors as input, the gradient is computed in the similar way by flattening and arranging the elements in the tensor in a vector form. Let $\mathbf{Y} = g(\mathbf{X})$, and $z = f(\mathbf{Y})$ are two functions taking tensors as input, the tensor version of chain rule is written as:

$$\nabla_{\mathbf{X}} z = \sum_j (\nabla_{\mathbf{X}} Y_j) \frac{\partial z}{\partial Y_j}. \tag{2.23}$$

Back-propagation in ANNs

A single layer in a neural network can be represented as a computation graph in Figure 2.10(a). Their gradients for all operation nodes are derived as shown in Figure 2.10(b). According to the chain rule of calculus, the partial derivative value between any two nodes equals to the multiplication of all partial derivative values on the route connecting the two nodes, e.g., in Figure 2.10(b).

$$\frac{\partial \hat{\boldsymbol{z}}}{\partial \boldsymbol{w}} = \frac{\partial \hat{\boldsymbol{z}}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{y}} \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{w}}. \tag{2.24}$$

In back-propagation, the topmost node is assigned with value 1, and its child nodes, for example $\hat{\boldsymbol{z}}$, stores the sum of multiplications between 1 and its own gradient $\frac{\partial \hat{\boldsymbol{z}}}{\partial \boldsymbol{z}}$. Then the next child nodes store their gradients that calculated similarly.

Now considering a supervised learning problem, in which loss is given by $L(\hat{\boldsymbol{y}}, \boldsymbol{y})$, solved by a multilayer fully connected neural network. The value of loss function depends on the output $\hat{\boldsymbol{y}}$ and the label $\boldsymbol{y}$. Suppose we have ℓ layers in the model, each layer consists of a linear transformation like $\boldsymbol{y} = \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}$ and a non-linearity function σ, the computational model can be expressed as Equation 2.25 :

$$\begin{aligned} &\boldsymbol{h}^{(0)} = \boldsymbol{x}, \\ &\boldsymbol{h}^{(i)} = \sigma(\boldsymbol{w}^{(i)} \boldsymbol{h}^{(i-1)} + \boldsymbol{b}^{(i)}), \quad (i = 1, \cdots, \ell). \end{aligned} \tag{2.25}$$

We can obtain the loss for some input $\boldsymbol{x}$ by performing a forward pass (or forward propagation) as shown in Figure 2.10(a). Then we can back-propagate the loss as shown in Figure 2.10(b).

2.3 Scene Observation and Camera Pose

In this part, we first introduce basics on camera pose and its relation to image content, and then the relationship between image content and different camera poses. We use "scene" to denote a place with a collection of tangible objects. The objects are assumed to be static. The imaging process is to generate their perspective projections on a 2-D image plane.

2.3.1 Pinhole Camera Model for Scene Observation

An imaging system generally consists of a "lighting" source and a "scene" that reflects or absorbs light energy. Images are records of light intensity measured by sensors. In the measuring process, the continuous image space is sampled and quantified to a set of discrete numbers. For more introduction about digital imaging and sensors we refer to the second chapter of the textbook *Digital Image Processing* (Gonzalez & Woods, 2002).

We use the simple pinhole camera model to represent an imaging process. A pinhole camera catches the rays from a 3-D point to its optical center and projects them as points on a 2-D image plane. Note only the points in front of the image plane are recorded. However, this fact is ignored in our discussion when we need an "ideal camera". Though simple, the model is sufficient to reveal the imaging geometry of perspective projection. Suppose there is a pinhole camera, whose optical center is at the origin of a "world" coordinate system and the principal axis point to the positive direction of Z-axis. An image plane located at $Z = f$ and perpendicular with Z-axis, f is the focal length. The image coordinate system locates at the interaction of the principal axis and the image plane and has an x-axis and y-axis, parallel with X-axis and Y-axis in the world coordinate system respectively. A point P with coordinates (X, Y, Z) is then projected onto the image plane at point p with the image coordinate (x, y). Figure 2.11 is a schematic drawing for imaging geometry of a pinhole camera model. The coordinates of a point on the image plane, corresponding to the point in the scene, is given by the property of similar triangles (Equation 2.26):

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}. \tag{2.26}$$

A scene to be imaged can be formally represented as a function S: $\mathbb{R}^3 \to \mathbb{R}^K$ producing a description vector $S(P)$ for each point P in the scene. The description vector includes various properties of the point, for example, color, transparency, material, etc. The image is also defined as a function I: $\mathbb{R}^2 \to \mathbb{R}^K$, which takes the description values $I(x, y)$ at each point x, y on the image plane. When taking an image, a point in a physical scene is either occluded or transformed as a ray from the point to the camera's optical center. Every observable point in the scene can be uniquely mapped to a point on a unit sphere that is centred at the optical center of the camera. More formally, the observable scene is a real projective space, denoted as RP^2. The sphere is a 2-D manifold embedded in 3-D

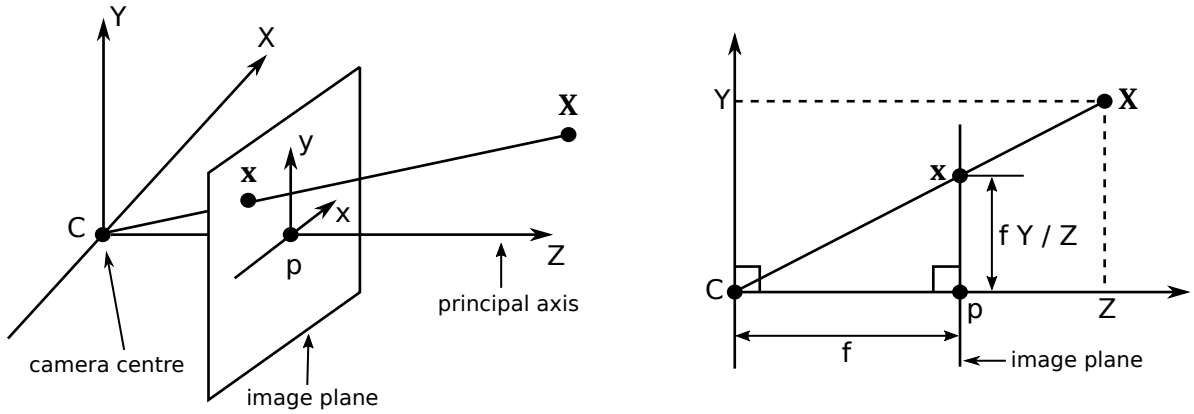


Figure 2.11: The camera geometry in viewer-centered coordinate system: the image plane locates at $Z = f$, where point $P(X, Y, Z)$ in physical scene falls onto the image plane at $p(x, y)$; C:optical centre of a camera; f:distance from image plane to optical centre.

Euclidean space and denoted as S^2. Let $\boldsymbol{m}$ be a unit vector representing a point on the unit sphere, the observable scene is then represented as a function $s(\boldsymbol{m})$. In a observer-centred reference frame, i.e., X, Y, Z - coordinate system, and the image plane at $Z = f$, a physical scene is a function $S(X, Y, Z)$ w.r.t points like $P(X, Y, Z)$. The homogeneous scene is $s(m_1, m_2, m_3)$, in which $\boldsymbol{m} = (m_1, m_2, m_3)^T$, and $\|\boldsymbol{m}\| = 1$. Let $I(x, y)$ denote a image function obtained in the scene, it is written as (Equation 2.27):

$$I(x, y) = s(m_1, m_2, m_3), \tag{2.27}$$

where $m_1 = \frac{x}{\sqrt{x^2+y^2+f^2}}, m_2 = \frac{y}{\sqrt{x^2+y^2+f^2}}, m_3 = \frac{f}{\sqrt{x^2+y^2+f^2}}$.

Let $\mathbf{\Pi}$ denotes the perspective projection operation, and the operation maps points in the 3-D physical scene onto the 2-D image plane: $\mathbf{\Pi} : \mathbb{R}^3 \mapsto \mathbb{R}^2$. Using homogeneous coordinates to represent 2-D points on the image plane, the operator implements perspective projection by matrix-vector multiplication

$$\mathbf{\Pi} : \boldsymbol{P} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \mathbf{K}\boldsymbol{P} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \\ 1 \end{pmatrix} = \boldsymbol{p}, \tag{2.28}$$

where $\mathbf{K}$ is the matrix form of intrinsic parameters of a camera and the last two steps are established in the homogeneous coordinates context. With consideration of the realistic situation where there are distortions, a more practical projection matrix is used:

$$\mathbf{K} = \begin{pmatrix} a_x \cdot f & s & t_x \\ 0 & a_y \cdot f & t_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.29}$$

The matrix in Equation 2.29 represents a full intrinsic calibration of a pinhole camera. The elements in the matrix are called intrinsic parameters of a camera, including the distorted focal length $a_x \cdot f$ and $a_y \cdot f$, the offset of the image origin from the optical axis intersection (t_x, t_y), and a skew parameter s.

2.3.2 Camera Pose Representations

We can represent a camera pose in terms of a rotation and a translation. The camera pose has 6 degrees of freedom (6DoF), 3 for translation and 3 for rotation. There are various methods to represent rotations, including rotation matrix, principle rotations, Euler angle, quaternions. A rotation representation with three additional parameters for translation can represent the full 6DoF camera pose. Here we introduce rotation matrix and quaternions as they are used in our experiment datasets. We refer the second part of the book *State Estimation For Robotics* for readers who need a comprehensive introduction of camera pose representation and relevant issues.

For a point (x, y) on 2D plane, we can easily write its counterpart (x', y') after rotating θ in the counter-clock wise direction about the origin, (2.30)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2.30}$$

The rotations that are restricted on a planar formed by two axes can be written as the Equation 2.33),

$$\mathbf{R_x}(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \tag{2.31}$$

$$\mathbf{R_y}(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \tag{2.32}$$

$$\mathbf{R_z}(\theta_3) = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

With predefined rotation order, rotations in 3D space can be expressed by a sequence of the above three principal rotations and implemented as matrix multiplication. One possible implementation is as the following:

$$\mathbf{R} = \mathbf{R}_z(\theta_3)\mathbf{R}_y(\theta_2)\mathbf{R}_x(\theta_1). \tag{2.34}$$

Unit quaternion is another rotation representation. Quaternions are generally represented in the form $w + p\boldsymbol{i} + q\boldsymbol{j} + r\boldsymbol{k}$, where w, p, q, r are real numbers, and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are three imaginary quaternion unit. According to Euler's rotation theorem, a rotation in 3D space can be equally represented by a rotation of an angle θ about a fixed axis. The rotation axis indicates a direction that can be represented by a unit vector $\boldsymbol{u} = u_x\boldsymbol{i} + u_y\boldsymbol{j} + u_z\boldsymbol{k}$. Then a point (v_x, v_y, v_z) in 3D space can be written in a pure quaternion as $\boldsymbol{v} = a_x\boldsymbol{i} + a_y\boldsymbol{j} + a_z\boldsymbol{k}$, a rotation of angle θ around the axis $\boldsymbol{u}$ to the point can be expressed as,

$$\boldsymbol{q} = \cos\frac{\theta}{2} + (u_x\boldsymbol{i} + u_y\boldsymbol{j} + u_z\boldsymbol{k})\sin\frac{\theta}{2}, \tag{2.35}$$

$$\boldsymbol{v}' = \boldsymbol{q}\boldsymbol{v}\boldsymbol{q}^{-1}, \tag{2.36}$$

where $\boldsymbol{v}'$ is the rotated vectoer and $\boldsymbol{q}^{-1} = \cos\frac{\theta}{2} - (u_x\boldsymbol{i} + u_y\boldsymbol{j} + u_z\boldsymbol{k})\sin\frac{\theta}{2}$. Two rotation quaternions $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ can be composed into one equivalent rotation $\boldsymbol{v}' = \boldsymbol{q}_2\boldsymbol{q}_1\boldsymbol{v}\boldsymbol{q}_1^{-1}\boldsymbol{q}_2^{-1}$ by quaternion's multiplication. Quaternion is a more compact rotation representation compare to rotation matrix, and consists of four unified parameter in the same space while the axis-angle representation is composed of three values from euclidean metric and one from angle which is more convenient for interpolation.

If a quaternion is represented by $w+p\boldsymbol{i}+q\boldsymbol{j}+r\boldsymbol{k}$, then the equivalent matrix, to represent the same rotation, is (2.37):

$$\mathbf{R} = \begin{bmatrix} 1-2*q^2-2*r^2 & 2*p*q-2*r*w & 2*p*r+2*q*w \\ 2*p*q+2*r*w & 1-2*p^2-2*r^2 & 2*q*r-2*p*w \\ 2*p*r-2*q*r+2*p*w & 1-2*p^2-2*q^2 & \end{bmatrix}. \tag{2.37}$$

2.3.3 SO(3) and SE(3)

In this part, we introduce two groups that are closely related to camera pose in terms of rotation and translation in 3D space. All possible rotations consist of a group, i.e., the **3-D rotation group**, denoted as $\boldsymbol{SO(3)}$. All 3-D translations and 3-D rotations compose a group, **3-D Euclidean group**, referred as $\boldsymbol{SE(3)}$. We can associate each element in SE(3) with a camera pose.

Group is defined as a set G together with an binary operation $\cdot$ that combines any two elements a and b to form another element c, i.e., $c = ab$, such that the combination $(G,)$ satisfies the four group axioms:

- **Closure**, for all a and b in G, the result of $a \cdot b$ is still in G;
- **Associativity**, for a, b and c in G, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$;

- **Identity element**, there exists one element e in G such that for any element a in G, $e \cdot a = a \cdot e = a$;

- **Inverse element**, for any element a in G there is another element b in G such that $a \cdot b = e$; b is the inverse of a and denoted as a^{-1}.

Writing a 3-D space point in vector form, the transformation is a linear mapping by multiplying a transformation matrix with the vector. We use $\mathbf{R}^{(n)}$, $\mathbf{T}^{(n)}$, $(n = 2, 3)$ to denote rotation matrix and translation matrix in 2-D and 3-D space respectively. When the context is clear, the dimension indicator is omitted. The rotation matrix in 3-D space is a 3 by 3 matrix and the point vector is written in form of $(x, y, z)^T$. Besides, $\mathbf{R}$ is an orthogonal matrix, i.e., $\mathbf{R}\mathbf{R}^T = 1$, *and* $det\,\mathbf{R} = 1$. We write a rotation group in matrix form as Equation 2.38:

$$SO(3) = \{\mathbf{R} \mid \mathbf{R} \in \mathbb{R}^{3\times 3},\ \mathbf{R}\mathbf{R}^T = 1\}, \quad \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{2.38}$$

The translation matrix in 3-D space is a 4 by 4 matrix, it is a identity rotation combined with a translation as Equation 2.39:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.39}$$

To be competitive to this matrix, the to be transformed vector is written in homogeneous form of $(x, y, z, 1)^T$. The combination of rotation and translation, namely, Euclidean transformation, is also a 4 by 4 matrix, which is the combination of $\mathbf{R}$ and $\mathbf{T}$. Let $\boldsymbol{v} = (t_x, t_y, t_z)^T$, the Euclidean transformation is written in the following form (Equation 2.40):

$$SE(3) = \{\mathbf{E} \mid \mathbf{E} = \begin{bmatrix} \mathbf{R} & \boldsymbol{v} \\ \mathbf{0} & 1 \end{bmatrix},\ \mathbf{R} \in SO(3)\ and\ \boldsymbol{v} \in \mathbb{R}^3\}. \tag{2.40}$$

When there is no need to specify details of a transformation matrix, we often use marks as g_R or g_E to represent a rotation with matrix $\mathbf{R}$ or an Euclidean transformation with matrix $\mathbf{E}$. Obviously, the multiplication of any two rotation matrix or two Euclidean matrix results a new rotation matrix or new Euclidean matrix. Note that $SO(3)$ and $SE(3)$ are **non-abelian group** or **non-commutative group**, which means that switch the order of two rotation or Euclidean transformation usually leads to different results. When a consecutive transformation is represented as a sequence of matrix multiplication, change the order of any matrix will also affects the final results.

2.3.4 Images with Different Camera Poses

When the perspective projection of a pinhole camera is introduced in the previous section, we use an observer-centred coordinate frame (camera frame). However, to represent 3D points in a scene a place-centred coordinate frame (world frame) is preferred. As moving an object with respect to the camera is symmetrically equal to moving the camera with respect to the object, we can derive the transformation from camera frame to world frame as a rotation and a translation (denoted as $[\mathbb{R}|\boldsymbol{t}]$). Then we can derive the relationship between a point on image plane in camera frame and its corresponding 3D scene point in world frame. Let $\boldsymbol{P}$ and $\boldsymbol{p}$ denote a 3D scene point and its projection on 2D image plane in camera frame, we have the relation $\boldsymbol{p} = \mathbf{K} \cdot \boldsymbol{P}$ as in Equation2.28, where $\mathbf{K}$ is the projection matrix. Let $\boldsymbol{P'}$ and $\boldsymbol{p'}$ denote the same 3D scene point and image point in the world frame, we can relate the 2D image point in camera frame with its 3D scene point in world frame as

$$\boldsymbol{p} = \mathbf{K}\,[\mathbf{R}|\boldsymbol{t}]^{-1}\,\boldsymbol{P'} \tag{2.41}$$

$$\hat{\boldsymbol{p}} = \mathbf{K}^{-1}\boldsymbol{p} = [\mathbf{R}|\boldsymbol{t}]^{-1}\,\boldsymbol{P'}, \tag{2.42}$$

where $\hat{\boldsymbol{p}}$ is called as a canonical point that can be seen as projected with an identity projection matrix (i.e., $\mathbf{K} = \mathbf{I}$) and $\mathbf{M}^{-1}$ represents the inverse matrix of $\mathbf{M}$. With the above equations, we can determine the location of an image point given a camera pose and the corresponding 3D scene point and thus the cotent of an image can be related to camera pose.

Chapter 3

Related Work to Place Learning

This chapter consists of three parts related to this work. In the first part, we draw a sketch evolution route on deep neural networks. Milestone architectures are mentioned and emphasized with the highlights on the ideas that enable deeper and more powerful network structures. In the second part, we introduce two generative models, i.e., variational autoencoders (VAEs) and generative adversarial networks (GANs), which are eye-catching research topics on deep learning in recent years. We also justify the reason for using VAE with stacked Conv / TransConv layers as the main network architecture in our experiments. In the last part, we review state of the art on visual place recognition. Despite existing reviews on this topic, such as Brejcha and Čadík (2017); Ji et al. (2015); Lowry et al. (2016). We put our efforts to reveal the influence of the rapidly developing deep learning on place recognition, camera pose estimation, and place representations.

3.1 The Evolution of Deep Convolutional Networks

The architecture of neural networks has a large impact on their performances in specific tasks, while there are no "golden rules" to follow. The initial design of a convolutional network consists of several stacked convolutional layers and fully connected subsequent layers to generate task-specific output, e.g., the probability values over the classes. LeNet 5, an early implementation of multilayer convolutional network that consists of 3 convolutional layers and two fully connected layers are used for handwrite number recognition. Nowadays, convolutional networks become deeper and deeper, and new functional components are designed to boost up the networks' performance. We list here some well-known network architectures, including AlexNet, GoogLeNet, VGGNet, ResNet.

AlexNet, developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, is the first popular convolutional network in Computer Vision, which has stimulated the deep learning study (Krizhevsky et al., 2012). The network took the first place in image classification task in the ImageNet challenge (ILSVRC) 2012 and outperformed other methods significantly in the competition. Russakovsky et al. (2015) report the competition results. AlexNet has

5 convolutional layers stacked on top of each other, followed by 3 fully connected layers (see Figure 3.1). This work also introduce "Dropout" and Local Response Normalization (LRN) as the means to prevent overfitting. Figure 3.1 illustrates the architecture of AlexNet. In the original implementation, the network is split into two subnetworks running on two GPUs to enable the training process.

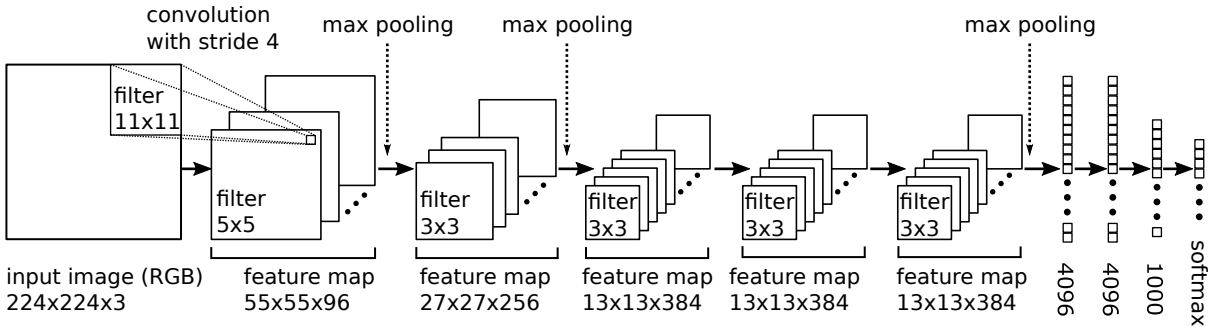


Figure 3.1: The AlexNet Architecture.

VGG-nets is proposed by Karen Simonyan and Andrew Zisserman in ILSVRC 2014 (Simonyan & Zisserman, 2014). VGG-nets has 4 blocks of stacked convolutional layers followed by a pooling layer showing that the depth of a network matters in performance. VGG16Net contains 13 3×3 convolutional layers, 5 2×2 max-pooling layers, and 3 fully connected layers. VGG19Net has 3 more convolutional layers. Figure 3.2 depict the VGG16Net and VGG19Net structure. The success of VGG-nets proved that deeper networks of stacked convolutional layers can improve the network's performance. However, VGG-nets is memory and training expensive because of its 140M parameters.

GoogLeNet, the ILSVRC 2014 winner, was proposed by Szegedy and his colleagues from Google (Szegedy et al., 2015). They introduced "Inception Module" and used average pooling instead of fully connected layers to reduce the number of parameters. In Figure 3.3 a), we depict a typical inception module, which takes advantage of convolutional layers with different receptive fields. Output modules are given in Figure 3.3 as well. Average pooling drastically reduces the number of parameters compares to fully connected layer while not harm the performance much. The total number of inception network is 4M (compared to AlexNet with 60M). There are also follow up versions of inception networks for better performance by modifying the Inception modules (Szegedy, Ioffe, et al., 2016; Szegedy, Vanhoucke, et al., 2016).

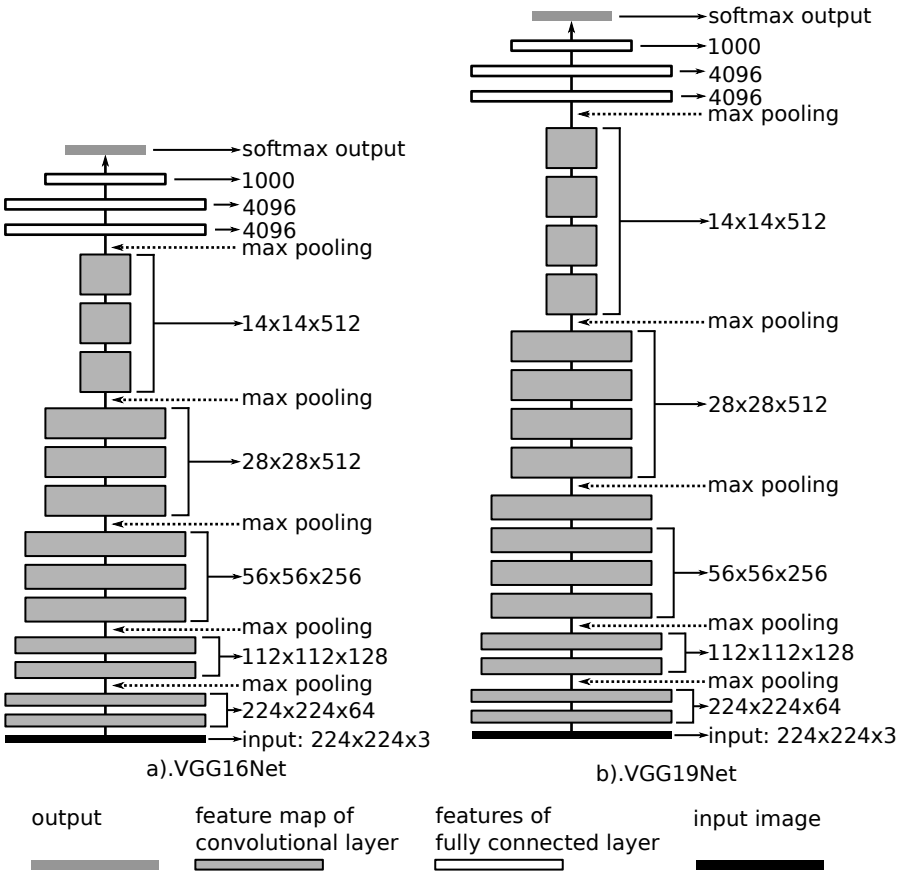


Figure 3.2: The VGGNet Architectures, a) VGG16Net, b)VGG19Net.

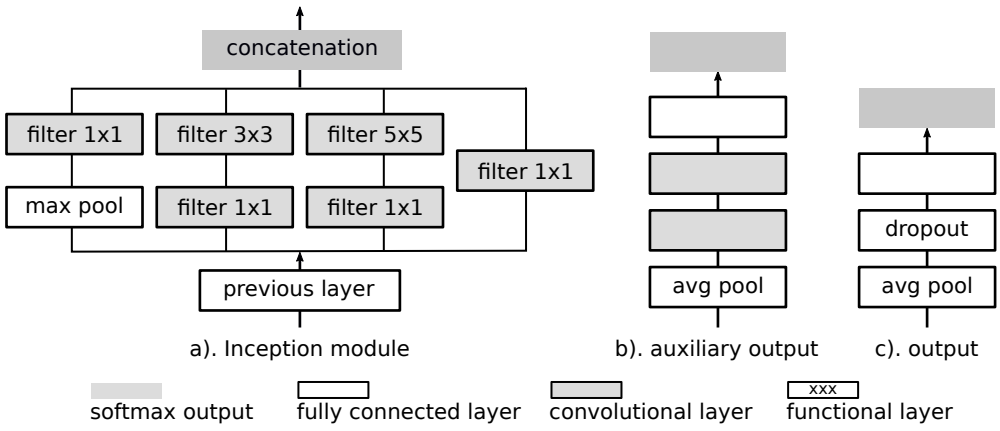


Figure 3.3: Inception Network, a). Inception Module, b). Auxiliary Output Module, c). Output Module.

ResNet, the winner of ILSVRC 2015 in classification task is developed by Kaiming He and his colleagues (K. He et al., 2016). The winner network achieves 152 layers in depth. ResNet is characterized by its skip connections, heavily used batch normalization, and no fully connected layer at the end of the network. A residual block (ResBlock) in ResNet consists of two convolutional layers and a skip connection that bridges the input to the output as shown in Figure 3.4. The authors proposed that the residual block which performs an identity mapping making training easier. Let $H(\boldsymbol{x})$ denotes the desired function that the block is supposed to learn, and the two convolutional layers learn a function $F(\boldsymbol{x})$. Without a by-pass connection, the two functions are identical, i.e., $H(\boldsymbol{x}) = F(\boldsymbol{x})$, while with the connection $H(\boldsymbol{x}) = F(\boldsymbol{x}) + x$, i.e., the convolutional layers learn a residual $F(\boldsymbol{x}) = H(\boldsymbol{x}) - x$.

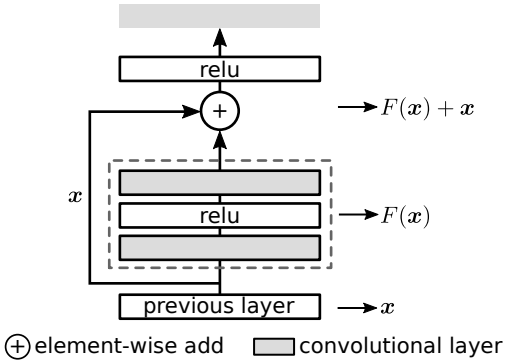


Figure 3.4: A ResNet Block.

As we summarize the development of deep convolutional networks, we show how network structures are evolving, first going deeper, and then more adding functional components. In our experiment, we begin with a structure with 5 Conv / TransConv layers in the encoder/decoder. In some experiments, we replace the usual convolutional layers with ResBlocks for better performance.

3.2 Generative Models and Representation Learning

Generative models are often used to learn representations that preserve the information of data distribution while revealing salient structures in the data distribution. Generative models are generally capable of unsupervised learning where labeling information is missing. However, to supports learning semantically meaningful representations, adding extra labeling information is also possible when it is available. The learned representations capture to some extent the intrinsic structure of the data distribution; on the other hand, one can use the learned properties to produce new data samples.

Generative models are usually related to the latent variable model, where the latent random variable is related to the data generation process. The goal of generative models is to learn the dependency structure of the latent space and the data. Generative adversarial networks and variational auto-encoder networks are two popular generative models in recent years. They share a common setting that both models employ neural networks to map a latent distribution $p(\boldsymbol{z})$ to a generated distribution $\hat{p}_{\boldsymbol{\theta}}(x)$ such that $\hat{p}_{\boldsymbol{\theta}}(x)$ matches the true data distribution $p(\boldsymbol{x})$. Though we focus on the VAE-based methods in our experiments, we still review both VAEs and GANs as we want to present a comprehensive image on representation learning to show their potential for place representation learning.

3.2.1 VAE: Variational AutoEncoder

VAE is a special type of autoencoders. An autoencoder is a parametric function (implemented as a neural network) to reconstruct its output from the input. Autoencoders usually share similar architecture, i.e., an encoder part that maps an input to a latent representation, while a decoder takes the latent representation to produce a reconstructed input. Figure 3.5 shows a typical autoencoder structure, where $z = f(x)$ is an encoder takes input x and produces its latent variable z, and a decoder $\hat{x} = g(z) = g(f(x))$ tries to reconstruct x from z. However, what matters is not to train an autoencoder to copy input perfectly to output but to care about whether the learned latent representation reveals some structural properties of the dataset.

VAE is an important unsupervised learning framework to learn complicated distributions (Kingma & Welling, 2013). Compare to typical autoencoders, VAE makes a strong assumption on the distribution of the latent variable, resulting in an additional regularization term in VAEs loss function, i.e., the KL-divergence between the assumed distribution and the encoded distribution. VAE uses variational approaches to learn the latent representation with the Stochastic Gradient Variational Bayes (SGVB) algorithm for training.

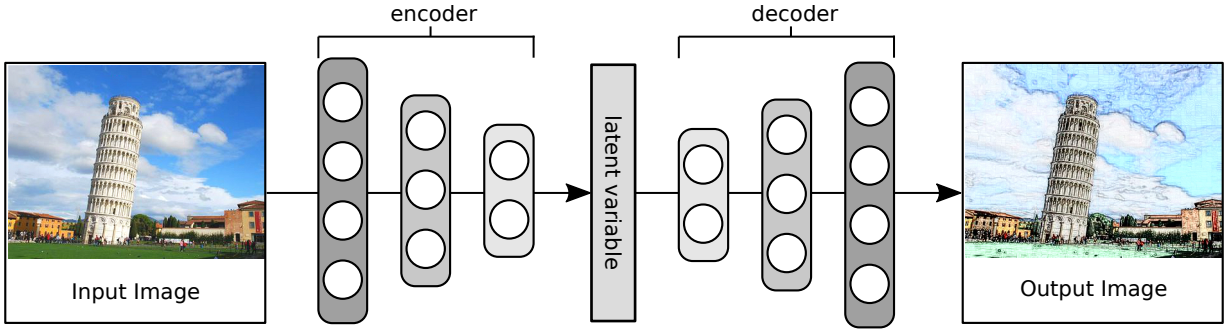


Figure 3.5: Schematic structure of an autoencoder, where encoder and decoder are implemented as neural networks.

There are conditional variations of VAE model when there are extra labels or other information y is available. According to different assumptions about the relationship between latent variable z, data sample x, and label y, there are different strategies to incorporate the condition into VAE. The variations of VAE show that the conditional generative model is not a fixed architecture but a rather flexible framework that we can incorporate extra information or labels according to different assumptions and problem scenes. In the place representation learning problem, the extra information can be categorical labels or camera poses. A simple way of extending VAE to a conditional version is to consider the conditional distribution $p(x|y)$. This version of conditional VAE has not much difference to the original non-conditional VAE and is also trainable with the similar reparameterization trick and back-propagation algorithm. In another conditional VAE, the condition y is assumed to be independent of the latent variable z, i.e., $p_\theta(z|y) = p_\theta(z)$. This type of conditional VAE is mentioned by Pandey and Dukkipati (2017). Sohn et al. (2015) focus on a label prediction problem rather than image generation. The proposed conditional VAE takes images as label information together with the latent variable in the generator to predict labels. In this case, the label information is treated as the data to be generated while the data x is treated as labels. Pandey and Dukkipati (2017) associate latent variable with the labels other than data samples. The model, on the one hand, ensures the latent variable containing structural features of the data distribution, on the other hand, makes the latent variable containing label information.

3.2.2 GAN: Generative Adversarial Networks

Generative Adversarial Network (GAN) is first proposed by Goodfellow and his colleagues (Goodfellow et al., 2014). It achieves impressive results in various tasks related to image generation, such as image generation (Radford et al., 2015), image to image transfer (Isola et al., 2016), text to image generation (Reed et al., 2016), and super-resolution image generation (Ledig et al., 2016).

The core idea of GAN can be summarized as a zero-sum game of two players, a generator and a discriminator. In this game, the generator tries to fool the discriminator with generated images, while the discriminator tries to distinguish between generated images and real data images. The training process can be described by an adversarial loss which consists of a generator loss and a discriminator loss for generator network and discriminator network in the actual training process accordingly. Optimizing the adversarial loss, on the one hand, increases the power of discriminator network to distinguish generated images from real ones, on the other hand, drives the generator network to produce images as close to the original images as possible. An ideal result for the generator is that the discriminator network cannot distinguish real and generated samples, and gives a 0.5 probability guess for both real and fake images.

Although the adversarial training process is clear and simple, optimization of GAN often leads to model collapse. Model collapse is a well-recognized problem in GAN training (Arjovsky & Bottou, 2017). A collapsed model always produces a single sample or a small set of similar images, which cover only a very limited portion of real data. Training stability is still an evolving research topic, while there are already various methods to address the problem. Salimans et al. (2016) suggest two methods to enforce the improvement of diversity, i.e., feature mapping and mini-batch discrimination. Feature mapping, instead of directly maximizing the output of discriminator, matches the generated samples with the statistics of real data, while mini-batch discrimination allows the discriminator to investigate the diversity from a batch of images. The Unrolled GAN (Metz et al., 2016) rolls the generator parameters back after several adversarial steps, which enables the generator to "predict" the future behavior of the discriminator. This operation prevents the generator from producing similar images. The "unroll" operation can be implemented in a fully differentiable way to perform back-propagation training. Mao et al. (2017) propose to use a least square loss function for the discriminator such that generated samples that not fit the real data distribution can be improved. Wasserstein GAN (WGAN) (Arjovsky et al., 2017) is another important improvement of vanilla GAN to address the stable training problem. WGAN uses Wasserstein distance as a better metric of two distributions than KL- divergence. In the training process, a weight clipping trick is applied to prevent large oscillations in weight values. Being aware of the fact that the weight clipping in discriminator can cause a negative effect on stability and performance, a follow-up paper introduces a gradient penalty to improve training WGAN (Gulrajani et al., 2017).

In GAN, both generator and discriminator can be implemented as neural networks. In DCGAN (deep convolutional GAN) (Radford et al., 2015), convolutional/transpose convolutional layers are used in generator and discriminator accordingly to take advantages of their abilities in handling image data. The authors also proposed several strategies to improve the stability, e.g. 1). to use convolutional layers with a larger stride instead of convolutional layer followed by a pooling layer, and remove fully connected layers, 2) applying batch normalization (see Ioffe and Szegedy 2015) in all convolutional layers, 3) use leaky ReLU activation in discriminator, and Tanh for the last layer of generator and ReLU for other layers. Figure 3.6 shows the generator network used in the DCGAN paper (Radford et al., 2015). In practice, it is still necessary to balance the learning rate for discriminator and generator carefully.

3.2.3 The Combination of VAE and GAN

There are research efforts on combining GANs and VAEs to take the advantages from both. Makhzani et al. (2015) introduce adversarial training to an autoencoder architecture to approximate an implicit prior on latent representation. The resulted adversarial auto-encoder (AAE) matches the learned distribution of latent variable to an arbitrarily defined prior. The model consists of three components, i.e., an encoder which maps an input data samples to a latent vector, a decoder that reconstructs an input given a latent vector; and

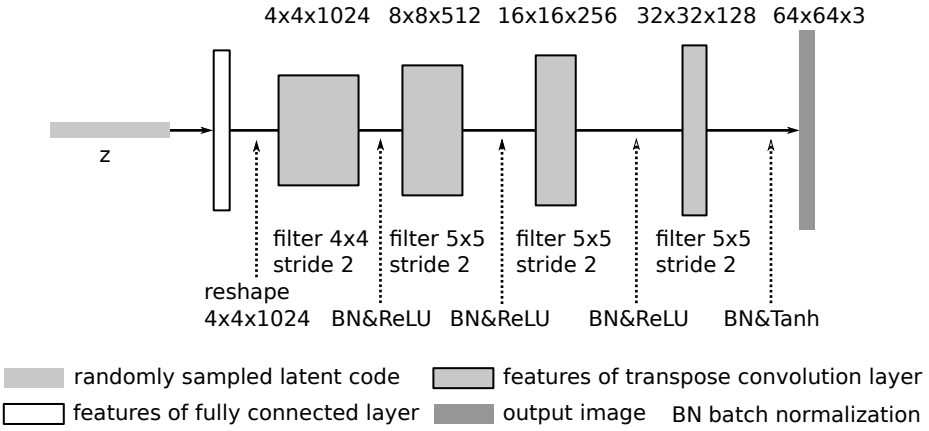


Figure 3.6: The generator network used in the original DCGAN paper.

a discriminator to distinguish between the encoded latent vector and the one sampled from the real prior. AAE has a similar encoding-decoding structure as VAE; however, instead of using KL-divergence as in VAE, the AAE employs adversarial learning to measure the difference between two distributions. Mescheder et al. proposed to use an adversarial metric to measure the difference between the prior distribution of the latent variable and the recognized distribution rather than the KL-divergence in vanilla VAE (Mescheder et al., 2017). A real value function is added to the VAE architecture as a discriminator to produce such measurement. Similar ideas appear in Ian Goodfellow's NIPS 2016 tutorial (Goodfellow, 2016). AAE and adversarial variational Bayes have a similar network structure but vary in their theoretical background. Both models try to find a better replacement to KL divergence (KL-divergence is not a metric since usually $D_{\mathrm{KL}}(P||Q) \neq D_{\mathrm{KL}}(Q||P)$), and the AAE model enforces the aggregated distribution $\int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})p_{\mathrm{data}}(\boldsymbol{x})\mathrm{d}x$ to approximate the prior $p(\boldsymbol{z})$ while in adversarial variational Bayes the authors used a learned metric to measure the difference between two distributions (Mescheder et al., 2017). Bidirectional GAN (Donahue et al., 2016) adds an encoder to GAN, and modifies the discriminator to take both latent variable and image data (including real data and generated data) as input, and thus enables the GAN to project data distribution back into the latent space.

There is also a conditional version of the combined model. The CVAE-GAN model is an example of (Bao et al., 2017). The proposed model has a classifier in parallel with discriminator to ensure the generated image is compatible with the given conditional information in the original dataset.

3.2.4 VAE for Visual Place Representation Learning

GANs and VAEs are widely investigated generative models in recent years. The core idea of both models share some common characters, i.e., build up the relationship between a prior distribution (usually a simple one, such as multi-dimensional normal distribution)

and a much more complex data distribution. To build such relationship, GANs begin with a simple distribution and try to find corresponding images in data distribution given samples in the simple distribution, while VAEs begin from data distribution and try to map dataset images to the simple distribution. By adding extra structure to a vanilla VAE or GAN, researchers also try to combine both directions. Compare to GANs, VAEs tend to generate blurry images. What causes such a phenomenon is not known yet, while there is one guess that it is the intrinsic effect of maximum likelihood to minimize $D_{\mathrm{KL}}(p_{\mathrm{data}}||p_{\mathrm{model}})$. We employ VAE in our work, as we are interested in learning a latent space with VAE, while GANs focus more on the generating part.

3.3 State of the Art of Place Recognition

Visual place recognition aims at recognizing geographic locations or places from an image or an image sequence (video). The place to be recognized can be either natural or artificial such as a mountain(see Baatz et al. 2012 and Saurer et al. 2016), a desert area (e.g., Tzeng et al. 2013), or an extensively explored urban and indoor environment. Place recognition can be performed at different granularity ranging from a very coarse level (e.g., in which continent, country, city) to a much finer level (e.g., in which community, street). In a coarse level place recognition method, the earth surface (or study area) is often divided into a collection of non-overlapped cells representing different places, and visual contents are associated with these cells so that an image is classified into one of these cells. We refer to such problem setting as discrete visual place recognition. Discrete visual place recognition is often modeled as a nearest neighborhood query problem or a classification problem. In the former problem setting, when a new query is given, a place recognition system should find out from its image dataset the geographically nearest ones. As a classification problem, the query image should be assigned to a discrete place among various places. Place recognition methods of estimating the exact coordinates (e.g., geographic lat/lon coordinates from GPS) and orientations are referred to as camera pose estimation. We find several definitions that address the characters of different place recognition granularities. Hays and Efros (2008) generalize visual place recognition as "...estimating a distribution over geographic locations from single image..."; Arandjelovic et al. (2016); Zamir and Shah (2014) treat place recognition as an image retrieval problem. Bansal et al. (2011) combine image retrieval techniques with geometric methods to get an exact camera geo-location.

3.3.1 Discrete Place Recognition

Place recognition with street-level images

In place recognition where geo-reference is not used, we can connect images to places where they are taken, such as continent, country, city, or neighboring community. In this context the geo-localization problem is thus equivalent to a classification problem, i.e., categorizing images into different places. At an abstract level where rigorous geo-reference is missing, space is divided by a set of images, and each image represents a

non-overlapped subspace. The distance between the two images is not necessarily equal to their geographical distance but measured by some similarity function. In this case, when a query image is given, its geo-location can be approximately described by known images which were taken in nearby places and stored in a database. When topologic information is added, the scattered images can be connected if there exists a direct path.

Volunteered images can contain highly localizable content of unique geographic information as well as noise and ambiguity images that are difficult to recognize in terms of geo-localization. For example, if an image contains a scene of sea and beach, it must be taken from the coastal area while an image of cat or dog even with GPS tag are hard to be geo-localized from the image content. Hays and Efros (2008) propose an Im2GPS method using a data-driven approach to learn image representations from a bunch of popular hand-crafted features for image-retrieval based place recognition. The authors investigate the relationship between the contents of volunteered images and their geo-location from a worldwide image dataset of 6 million GPS-tagged images from Flickr. In the follow-up work, Hays and Efros (2015) further improve their "Im2GPS" method by introducing local descriptors (e.g., local color and texture histogram, a bag of SIFT words) and applied a learning method that combines KNN (k-nearest neighbors) and SVM (support vector machine) to refine the retrieval result. In the deep learning era, Im2GPS is enhanced by a deep neural network for matching visual features to discrete places (Vo et al., 2017). The authors model the place recognition as an image classification problem where query images are assigned to a cell of earth surface tessellation, however in the test time they turn to the original Im2GPS method and replace the original hand-crafted features with the deeply learned features. Regarding the temporal properties of an image that are uploaded by the same user, Kalogerakis et al. (2009) improve the geo-localization by integrating traveling trajectories of the photographers. Being aware of connections between user tags and image content. Gallagher et al. (2009) enrich the depiction of geolocation by adding semantic information from user tags to visual content to of an image.

Associating volunteered images with landmarks is a special type of place recognition. In both human and machine cognition, landmarks themselves provide strong visual and/or semantic clues about places. Y. Li et al. (2009) represent a related work driven by a massive amount of Flicker images. Their dataset consists of 30 million images with nearly 2 million of which labeled into one of 500 landmarks. The authors use a multi-class SVM to learn vectorized interest point descriptors as features to represent these images. Zheng et al. (2009) implement a web-scale landmark recognition engine that learns from 20 million GPS-tagged photos and online tour guide web pages resulting in visual models for 5312 landmarks from 1259 cities in 144 countries. To investigate visual localization and landmark identification on mobile devices, D. M. Chen et al. (2011) collect a set of 1.7 million images with ground-truth labels, geo-tags, and calibration, as well as a set of cell phone query images in San Francisco. This dataset is collected by a mobile mapping vehicle equipped with various surveying and imaging sensors, such as camera, GPS, IMU. For

place recognition, a vocabulary tree of SIFT descriptors is used to characterize places with geometric verification. Zamir and Shah (2010) collect about 100,000 GPS-tagged images downloaded from Google Maps Street View for Pittsburgh, PA, and Orlando, FL. They use SIFT descriptors to represent 360-degree panorama images and a nearest-neighbor tree search pruning to find nearby images. Murillo et al. (2013) use Gist descriptor to represent panoramic images for place recognition. Gist, in contrast to local visual features, e.g., SIFT, is a global descriptor that characterizes the visual property of the whole image (Oliva & Torralba, 2001).

Before the rise of deep learning, manually designed features are widely used to characterize the visual property of places in place recognition research. These features bring visual place recognition to a certain level of success. With the amazing success of deep learning in computer vision tasks, applying deep learning to place recognition is getting more and more attention.

In the early stage when deep networks are introduced to handle place recognition problem, pre-trained networks (usually trained on large scale datasets, e.g., ImageNet) are used to extract deep features as place description. The reason is the pre-trained networks though not specifically trained on place recognition task, capture some common visual features from a large number of images. Z. Chen et al. (2014) report the usage of pre-trained Overfeat for place recognition. Overfeat is a network that performs object detection, localization, and classification tasks all into one. However, the generalization ability of their method is questionable as they manually select layer output of the best performance. Suñderhauf and his colleagues improve the performance of a pre-trained CNN in place recognition by forcing the network to work on image patches where salient landmarks are recorded (Sünderhauf, Shirazi, Dayoub, et al., 2015; Sünderhauf, Shirazi, Jacobson, et al., 2015). Their experiments reveal that features produced by pre-trained ConvNets are robust to viewpoint variance and conditional variance.

PlaNet (Weyand et al., 2016) can be viewed as a deep learning version of Im2GPS. PlaNet tackles place recognition a classification problem by subdividing the earth surface into thousands of multi-scale geographic cells. Its core component is a deep network that is trained with millions of geo-tagged images. The deep network is modified from the Inception network with batch normalization and a softmax layer generating the probability distribution of a query image for all geographic cells. The authors also employ a long short term memory (LSTM) architecture to enable the model to handle temporal coherence to geo-locate uncertain photos.

The emergence of institutional collections of huge amount of geo-tagged images, such as Google Street View, has fueled deep learning methods. These datasets are usually maintained by institutions and have geo-tags for supervised learning. Arandjelovic et al. (2016)

proposed a learning version of VLAD (Vector of Locally Aggregated Descriptors) to represent images for place recognition. The authors tackle street-level image geo-localization as an image retrieval problem. The proposed NetVLAD layer can be integrated into a neural network to facilitate learning with a weakly supervised ranking loss. The resulted network maps an image into a much lower dimension as NetVLAD vector, such that geographically closed images in this feature space are close to each other as well. They tested their method on Google Street View Time Machine dataset.

Cross-view place recognition

There are also researches about integrating multiple data sources for visual place recognition, aerial images, for example, represent another important geo-locational information source besides the street level images. This type of work is often referred to as cross-view place recognition. Bansal et al. (2011) propose to geo-localize a street-view image by matching building faade in that image with aerial images. If the correspondence of the oblique aerial and ground image is established, the camera pose where the street level image is taken can be calculated by geometric methods. Lin et al. (2013) propose to learn the correlation of two appearances from ground level images and airborne images with additional land cover information for geo-localization. Their method discriminates hand-crafted features from ground-level images with the features from image patches taken from aerial images and land cover map in a sliding window way. The authors built up a dataset of 6756 ground-level images from Panoramio, 182988 aerial images from Bing Maps, and land cover attribute from USGS GAP Land Cover Data Set. Castaldo et al. (2015) match street-level images and a GIS map by semantic segmenting the image and find the correspondence between the contact region of semantic patches and their counterpart features in the map.

The cross-view place recognition requires to compare two images from different viewing perspectives, which is the application scenario of Siamese networks (Bromley et al., 1993). Lin et al. (2015) propose a Siamese-style network named "Where-CNN" which learns to match ground-level images and aerial image patches with a margin loss. To test their method, the authors built up a dataset of 78,000 aligned cross-view image pairs: they collected Google Street View and 45-degree aerial view images from seven cities San Francisco, San Diego, Chicago, and Charleston in the United States as well as Tokyo, Rome, and Lyon including both urban area and suburban area.

Place recognition in SLAM

Besides estimating a robots location, visual place recognition has an important application as "loop closure" in, such that a robot (or other autonomous systems) can "remember" the place where it has already visited and thus generate accurate maps. The visual SLAM literature frequently addresses visual place recognition in outdoor, indoor, and smaller-scale environments using image or image sequence. In the comprehensive review of the

progress in SLAM from Cadena et al. (2016), the authors reveal the importance of visual and semantic representations for mapping and mention deep learning as an important technique that may bring significant changes to SLAM. Lowry et al. (2016) examine existing works according to the 3 major components of a visual place recognition system, i.e., the image processing module, the map, the belief generation module.

In many conventional visual place recognition systems, hand-crafted visual features and place description methods (such as the bag of visual words, BoVW) are their backbones (Sivic & Zisserman, 2003). Newman and Kin Ho (2005) illustrate how visually salient and wide-baseline stable visual features help in loop-closure. Sünderhauf and Protzel (2011) combine BRIEF, a local descriptor, and Gist, a global descriptor to encode places for loop closure. Angeli et al. (2008) extend the BoVW method to incrementally encode SIFT features and local color histograms for place description and use Bayesian filtering to estimate the loop-closure probability. FAB-MAP (Cummins & Newman, 2009, 2011) provides a highly efficient probabilistic SLAM system with visual place recognition and BoVW techniques. Nicosevici and Garcia (2012) extend BoVW method with an incremental vocabulary building process. The visual vocabularies are built online as more images are observed, new elementary clusters are extracted and added to the vocabulary and the complete set of clusters gradually converges. F. Li and Košecká (2006) select a set of discriminative SIFT features for characterizing individual locations to speed up recognition while keeping a high recognition rate. Gálvez-López and Tardós (2011) use FAST keypoints and BRIEF descriptors in place recognition to improve the computational efficiency of a SLAM system. In addition to point features, edge features are also used for place recognition in SLAM (Eade & Drummond, 2009).

Images that are consecutively captured in a short time interval are more likely recording the same place. The SeqSLAM system matches local query image sequences to a database of image sequences in place recognition with the above hypothesis (Milford & Wyeth, 2012). Similar efforts include Sequence Matching Across Route Traversals (SMART) (Pepperell et al., 2014) and the work of Hansen and Browning (2014), where a Hidden Markov Model (HMM) is used to capture the temporal property in image sequences.

Beside temporal property, additional spatial constraints are also used in the following two examples. Cadena et al. (2012) add geometric verification to BoVW within a Conditional Random Fields (CRF) model and use a novel normalized similarity score to measure the similarity of recent images in the observed image sequence. Stumm et al. (2013) introduce co-visibility graph for mutually visible landmarks (distinct visual features), such that the spatial correlations can be used in a generative place recognition algorithm.

3.3.2 Camera Pose Estimation

Knowing precise position gets more attention in robotics, and autonomous driving (Levinson & Thrun, 2010). Though various devices have GPS chips embedded, such as camera, mobile phone, and vehicle navigation device, the positions that are given by GPS can be noisy and uncertain (Zamir & Shah, 2014). The vision-based positioning system can be seen as a complementary or alternative solution to GPS-based localization: 1). GPS is limited to access in some urban area where dense buildings will block GPS signal; 2). Vision-based localization and pose recognition provide finer and richer information than the GPS signal, which is required by applications such as autonomous robots and vehicles.

Geometry-based camera pose estimation

Structure from Motion (SfM) techniques (see Szeliski and Kang 1994) for an example, are able to simultaneously reconstruct a 3D scene structure and camera pose from a set of feature correspondences from observed images. Estimating the absolute pose of a given query image with the help of such a pre-computed 3D model is often referred to as structure-based localization. Nowadays, SfM approaches are able to reconstruct a large scale 3D model from unordered images collected from the Internet and establish correspondences between image features and 3D scene points. Here we show some examples. Snavely and his colleagues present a framework for the creation of efficient correspondence and 3D reconstruction for extremely large image data sets (Snavely et al., 2006, 2008). Agarwal et al. (2009) present a system to match and reconstruct 3D scenes from a huge number of photos to build a 3D point cloud model for city Rome. These photos are collected from the Internet without knowing the calibration parameters of cameras. Heinly et al. (2015) represent a stream computing paradigm to construct a world-scale 3D model from unordered Internet image photo collections.

To reliably estimate the camera pose of a query image with a 3D point cloud of a scene, we need to know the correspondences between 2D features and 3D points. Once correspondences are found the camera pose can be estimated with high precision by geometric methods. These methods are well explained in the textbook by R. Hartley and Zisserman (2004). Consequently, the structure-based localization can be reduced to a problem of searching correspondences between 2D features on the query image and 3D scene points. The searching methods can be divided into approaches based on image retrieval and those based on direct matching. Image retrieval based methods first search through the database for similar images to the query image, then 2D-to-3D correspondences can be established by matching features in the query image against those in the database images of which correspondence of 3D points has been established. Direct matching approaches skip searching the intermediate database images and instead directly match features in the query image against 3D scene points.

Irschara et al. (2009) build a 3D model with a collection of images by SfM and link the point features on an image to their corresponding 3D point in the model. The vocabulary tree-based indexing is used to retrieve relevant images of a query image from the database where k highest ranked images are selected for feature matching. The authors apply a regular SIFT feature matching to find correspondences between 3D points and the query image features followed by pose estimation. In Prioritized Feature Matching (Y. Li et al., 2010), place recognition is performed in an opposite way by matching a representative set of SIFT features covering a large scene to a query image. Following heuristic searching strategies, this method searches the representative features from a visibility graph where each scene feature is matched to a set of images that contain the corresponding image feature. (Sattler et al., 2011) show that a vocabulary tree-based approximate search through all point descriptors for direct 2D-to-3D matching achieves higher registration rate, but it is still too computationally expensive to run in real time. Sattler and his colleagues also propose an active correspondence search where the spatial property of the 3D points is considered (Sattler, Leibe, & Kobbelt, 2012). If a 3D point has its corresponding feature found in the query image, its neighboring points are also visible in the image. The active search framework first identifies a 2D-to-3D correspondence between an image feature and a scene point, then selects the neighboring area around the point as candidates for 3D-to-2D matching. Y. Li et al. (2016) address the global pose estimation problem with 3D point cloud data and image features. They employ a direct bidirectional matching schema (first 2D-to-3D, then 3D-to-2D) to establish the correspondence between image features and 3D scene points in a large point cloud database covering many places around the world. To find better correspondences in image matching, Zamir and Shah (2014) refine multiple potential matches by selecting a single nearest neighbor for each query feature such that all matches are globally consistent. Direct matching methods generally outperform image retrieval methods which take intermediate images in the feature matching step, while direct matching methods require high resource demand when the database is large. Sattler, Weyand, et al. (2012) identify false positive votes as the major cause of the performance gap between the retrieval method and the direct matching, and propose a retrieval method that relies on a selective voting scheme to fill the gap. Their results show an improvement in comparison with image retrieval methods and direct matching methods.

Considering the scalability when matching through a large image dataset, researchers also try to develop compressed scene representations that leverage accuracy and efficiency. Sattler et al. (2015) compress a 3D model with hyper-points and fine visual vocabulary. A hyper-point records a set of locally unique matches between 3D scene points and an image feature to reduce the matching ambiguity. Bearing in mind that there are similar images of the same place with similar viewpoints in a large image dataset, Johns and Yang (2011) suggest compressing the dataset by clustering similar images to differentiate places and learn discriminative features for each place. Their method results in a set of place descriptions consisting of visual words and thus the place recognition problem can be addressed as matching features from the query image and the features of a place

description.

In addition to point cloud models, researchers also try to make use of other models. In the urban geo-localization scene, Bansal and Daniilidis (2014) extract 3D point-ray features from Digital Elevation Map (DEM) to compute correspondences between building corners in the DEM and a query image. Since the 3D models are usually constructed using SfM methods from images, Sattler et al. (2017) even argue that 3D models can be omitted in camera pose estimation. Their method illustrates that images with approximate geo-locations suffice to estimate precise camera pose.

Usually, the camera pose estimation aims at calculating the full 6 DoF camera pose, including 3 degrees for translation and 3 degrees for rotation (See Section 2.3). There are also efforts on reduced camera pose estimation problem. Armagan et al. (2017) estimate a reduced camera pose in 3 DoF, in which two for horizontal location, and one for direction from a coarse GPS position, by matching and recovering geometric correspondence between semantic features in the observed image and 2.5D models of the surrounding buildings. Baatz et al. (2010) reduce camera pose estimation to a pure homothetic problem by exploring the geometric constraints from vanishing points to remove 3D rotation. Košecká and Zhang (2002) also solve a reduced camera pose estimation problem by only considering the cameras orientation. They observed that in an indoor environment, lines are aligned with the principal orthogonal directions of the world coordinate frame when projected onto an image and impose strong constraints on camera pose in the scene.

Learning-based camera pose estimation

Undoubtedly, the geometric methods that reveal the geometric correspondence between a 3D scene point and the 2D image point are reasonable and precise in camera pose estimation. However, they suffer from inaccurate point feature extraction and correspondence searching. Moreover, they usually employ hand-crafted point features which can hardly represent semantic information. With the rise of deep learning, more efforts are being made to improve part or the whole process of camera pose recovery by embedding or utilizing deep networks in the pipeline. We roughly divided the efforts of embedding deep networks in camera pose estimation pipeline into three categories, 1) employing deep networks to replace part of the pipeline where conventional methods meet their performance bottlenecks, such as feature selection and correspondence matching; 2) building end-to-end architecture to regress absolute camera pose directly; 3) building end-to-end networks to incorporate geometric information in training to estimate relative camera pose.

DeTone et al. (2017) still follow the geometry-based pose estimation pipeline, i.e., first detecting point feature correspondence and then recovering geometric relationship. The whole system is a cascaded network architecture which consists of three sub-networks, i.e., two point-extractors to detect salient corner points from two consecutive images, and a

wrapper net to match corresponding points by calculating a homography matrix. The model is trained and tested on synthesized 3D models and 30 Static Corners Dataset and achieves significant improvements in corner point detection compare to hand-crafted detectors, and stronger robustness when the translation and rotation is large. RANSAC (random sample consensus) is a widely used algorithm to exclude the mismatched 2D-3D visual features in pose estimation. Brachmann et al. (2017) propose a differentiable RANSAC algorithm by introducing two learning-based versions for sample selection in RANSAC as 1). a softmax assignment, 2) stochastic expectation calculation. The proposed method jointly trains a sampling network and a scoring network to minimize the difference between generated matches and the optimal matches, such that the generated hypothesis leads to a robust camera pose. Zamir et al. (2016) show a multi-task network to perform wide-based line feature matching and camera pose estimation. The features are image patches of the same points and remain invariant under viewpoint change. The tasks include scene layout, object pose estimation, and surface norm estimation.

Kendall et al. (2016) made the first attempt to use deep learning method for camera localization. The PoseNet model is modified from the Inception network by replacing the softmax layer with pose regression layer to produce camera poses in quaternion form. In the following work, the authors investigate the pose uncertainty by using Bayesian ConvNets instead of the original PoseNet (Kendall & Cipolla, 2016). In another relevant paper from Kendall and Cipolla (2017), the authors compared the original PoseNet loss and two newly proposed loss functions. Since the translation and rotation are represented in two different metric space, the original loss function uses a beta factor to balance the two measurements. The first proposed loss function is a Laplace likelihood loss which involves two homoscedastic uncertainties for translation and rotation accordingly. The two parameters are learned during training. The second loss function is a geometric loss that uses directly projection error, which measures the distance of an image point that projected from a 3D point in the scene using estimated pose and the ground truth. This loss function requires a 3D model to provide the ground truth positions on image plane when performing re-projection.

In a series of works that follow PoseNet, 7-Scenes dataset[1] (Glocker et al., 2013) and Cambridge Landmarks dataset[2] are extensively used for testing and evaluation among different methods. Besides, TUM indoor dataset[3] and some SLAM dataset such as Oxford RobotCar dataset[4] are also widely used benchmark datasets in similar researches. Such dataset usually contains images and their ground truth camera poses or positions collected from indoor or outdoor scenes.

[1]MS 7-Scenes dataset https://www.microsoft.com/en-us/research/project/rgb-d-dataset-7-scenes/ access date 22.11.2018

[2]Cambridge Landmarks dataset http://mi.eng.cam.ac.uk/projects/relocalisation/ access date 22.11.2018

[3]TUM indoor dataset. https://hazirbas.com/datasets/tum-lsi/ access date 22.11.2018

[4]Oxford RobotCar dataset. http://robotcar-dataset.robots.ox.ac.uk/ access date 22.11.2018

Walch et al. (2017) use LSTM (long short-term memory) unit to further refine the high dimensional output from the first fully connected layer and regress translation and rotation separately. They test their method on TUM indoor dataset and Cambridge Landmarks dataset and show an improved result as compared to the original PoseNet. Melekhov et al. (2017a) propose an encoder-decoder CNN architecture with short-cut connections between encoder and decoder layers for estimating camera pose from a single RGB-image. They name this model as Hourglass Network. Their network is tested on the 7-scenes dataset and achieves better performance than PoseNet and LSTM-Pose. Motivated by multi-task learning modality, Wu et al. (n.d.) suggest to use two branches to regress the translation and the rotation separately, thus modify PoseNet by replacing the higher layers (near to the output) with two branch nets. They also use sine and cosine values of Euler angles to represent rotation. A data augmentation method named pose synthesis is designed to produce more training data that covers more pose space to cope with overfitting in training.

Researchers also use deep learning methods to estimate the relative pose from consecutive images by capturing the spatial-temporal correspondence between images. Melekhov et al. (2017b) estimate relative camera pose with Siamese-style network. The network consists of pre-trained AlexNet with a spatial pyramid pooling (SPP) layer instead of a common pooling layer. The authors trained this model on five landmarks image data (incl. Montreal Notre Dame, Piccadilly, Roman Forum, Vienna Cathedral, and Gendarmenmarkt) covering altogether 581,000 image pairs and validated with a subset of Yorkminster covering 22,000 image pairs. The authors also use DTU dataset[5] to evaluate the proposed method and compare with the hand-crafted point-feature based methods. The results show that their network structure achieves better accuracy compare to baseline CNN and geometry-based methods. Laskar et al. (2017) also investigate relative pose estimation by combining image retrieval and pose regression to make use of an existing image with known pose to predict the relative pose between a query image and its reference image. A Siamese-style network is first trained on the regression task and then fine-tuned for the retrieval task. In the inference stage, the network retrieves Top-K nearest images to query image and then a RANSAC-based method is used to filter out outliers and generate a final pose. Their method is tested on two indoor datasets, i.e., the 7-Scenes dataset and 5 scenes in a university building. Clark et al. (2017) graft a bi-directional LSTM unit on an Inception network to capture the correspondence. The model has been evaluated on 7-Scenes dataset and Oxford RobotCar dataset with an improved performance comparing to PoseNet.

Recently end-to-end deep learning based methods have employed geometric cues as supervision signals in relative pose estimation, including depth and optic flow. Depth and optic flow provide extra information other than color information in images. Costante et al. (2016) investigate the dense optical flow as a geometric cue in the relative pose estimation task. Motivated by researches in view synthesis such as DeepStereo (Flynn et al.,

[5]DTU dataset is collected in the section for Image Analysis and Computer Graphics at the Technical University of Denmark http://roboimagedata.compute.dtu.dk/ access date 10.07.2018

2015), a model which takes depth as intermediate information to interpolate views under a new camera pose, T. Zhou et al. (2017) use depth as intermediate information for distilling geometric structure from images. Their model consists of a single-view depth prediction network, which takes a single monocular image as input and predicts a depth map of the image as output, and a multi-view pose estimation network that estimates relative camera poses from already observed images. The loss function consists of the differences of depth between nearby warping views and the target image as well as pose. The warping views are calculated by image warping proposed in STN (spatial transformer network). The two functional networks are coupled by the loss during training but are decoupled at test time. The per-pixel loss is also weighted by a soft mask to strengthen the pixels that are more likely to be successfully modeled.

SfM-Net reported by Vijayanarasimhan et al. (2017) is another model that uses geometric cues to estimate the relative camera pose in terms of scene and object depth, camera motion and 3D object rotations and translations. The proposed network is similar to that in (Zhou et al. 2017) but designed for multiple tasks including optical flow prediction, depth prediction, camera motion estimation, object detection, and point cloud generation. DeMoN, reported by Ummenhofer et al. (2017), is a convolutional network trained end-to-end for depth, camera motion, surface normal, and optical flow estimation from successive unconstrained image pairs. The architecture is composed of multiple stacked encoder-decoder networks, the core part being an iterative network that is able to improve its predictions. The loss function consists of multiple terms of point-wise depth loss, motion loss, scale-invariant gradient loss. The method is tested on multiple datasets of indoor scenes. MapNet from Brahmbhatt et al. (2017) is a model that jointly performs relative pose estimation and global pose estimation. Valada et al. (2018) report a similar model "VLocNet", which estimates global pose and relative pose simultaneously from consecutive monocular images. The network architecture consists of a global pose regression sub-network and a Siamese-style sub-network for the relative pose estimation with parameter sharing trained with a Laplace likelihood geometric loss (Kendall & Cipolla, 2017). This network significantly outperforms other learning-based methods including PoseNet, Bayesian PoseNet, LSTM-Pose, VidLoc, Hourglass Pose, BranchNet, PoseNet-with geometric loss, and NNnet on the 7-Scenes dataset and Cambridge Landmarks dataset. VLocNet++, reported by Radwan et al. (2018) extends the VLocNet for multi-tasks of semantics segmentation, global pose regressing, and relative pose estimation. GeoNet also takes depth, optical flow, and camera motion as geometric cues (Yin & Shi, 2018). The geometric relationships are extracted over the predictions of individual modules and then combined as image reconstruction loss for static and dynamic scene parts separately. The adaptive geometric consistency loss increases the robustness towards outliers and non-Lambertian regions and is able to resolve occlusions and texture ambiguities.

Connecting visual place representation with camera pose

Buildings, bridges, the street signs, the topographical features of the terrain, all these objects in image observations contain rich semantic information. They can serve as strong clues to support place recognition or geo-localization. However, there are more generic scenes that may not leave distinct landmarks on image observations. For example, an image of an open office with wooden tables and computers may locate in San Francisco, Beijing, Berlin, or other cities. Although these images must contain certain visual features that are strongly correlated with a geographic place, the relationship is not strong enough to specifically pinpoint their positions. To precisely localize an image, visual features must be used with the combination of spatial information.

In discrete place recognition, spatial properties, specifically camera pose, are essential to visual place recognition as in most cases visual features along sometimes are insufficient. For example, the geometric relationship between two images can be used to verify the retrieved images among image retrieval based place recognition methods. In learning based methods, such as NetVlad or im2gps, the geographic locations of images are used to provide supervision in training.

Manually designed visual features play an important role in geometric methods for camera pose estimation. However, these features are insufficient to capture geographic property of a place at a high abstraction level. The learned visual features for camera pose estimation are implicitly bundled with certain to geographic locations in neural networks. The implicit mapping makes it difficult to interpret.

Since both visual and spatial information of place is important to place recognition, a specific mechanism to connect them can be beneficial for this task.

Chapter 4

An Approach of Learning Collective Place Definition

The rapidly increasing geo-data and fast forwarding techniques enable a data-driven approach to study the concept of place. In this Chapter, we begin from the attempts in geography and GIScience to understand place, and then derive three approaches for a collective place definition in the context of big geo-data and data driven-methods. Based on this approach, we propose a probabilistic place definition and introduce two deep learning techniques, i.e., VAE and comparative learning methods to approach this definition. Three computational models are proposed from the two deep learning techniques addressing different concerns about learning the place representations.

4.1 Towards the Collective Place Definition by Comparison

In this part, we first introduce a general framework in GIScience for place definition. Then we derive our specific approaches for the collective definition of place from a data-driven perspective. We emphasize the importance of comparison in our method.

4.1.1 A Comparative Approach for Place Definition

Modeling place in a digital world is identified as an essential task in GISystem and GIScience, due to the inherent vagueness and subjectivity of a place is often incompatible with the strict binary digital representation (Goodchild, 2011). More specifically, "the lack of precise locations, crisp boundaries and single universal names for many places that people talk about in everyday life" rises the main challenge of modeling a place (Davies et al., 2009, p. 175). Intuitively, the concept of place develops from space, where both objective geographic entities and subjective human activities, perceptions, and experiences settle. Based on the idea that place is anchored by "location", either the tangible or intangible entities can be assigned with locations in space with a spatial reference frame.

Thus the definition of place is closely coupled with the entities locations. However, the temporal, even spatial change of entities' locations often harms the purely spatial definition of place. Kabachnik (2012) provides an example of how the concept of place is shaped by mobility and settlement. He argues for disentangling the spatial component from place and considering a temporal component in place definition as well. Golledge suggests that "regardless of whether the tangible or intangible position is taken with respect to examining the sense of place, it should be possible to develop either a subjective or an objective scale (or some combination of the two) that captures the essence of a place" (Golledge & Stimson, 1997, p. 417). Cresswell constructs a framework of defining place by reviewing the genealogy of place in different disciplines. The framework composes multiple perspectives because "place stands for both an object (a thing that geographers and others look at, research and write about) and a way of looking" (Cresswell, 2014, p. 15). In the framework, the author identifies three levels at which geographers can approach the notion of place (Cresswell, 2014, p. 51), namely,

- a descriptive approach to place the ideographic approach concerns the distinctiveness and particularity of places;
- a social constructionist approach of place this approach focuses the unique attribute of a place in some specific social processes; and,
- a phenomenological approach of place this approach emphasizes the human experience of "in-place".

The three levels represent the level of depth in understanding places. They are not discrete methods but overlaps with each other.

With the context of big geo-data and data-driven research methods, we can derive a three-folds-approach accordingly from the general framework to achieve a collective place definition,

- *The constructive approach.* The constructive approach suggests the concept of place is constructed in a collective way by human generated data. No matter the users intentionally or unintentionally attend this process, their postings or digital footprints all contribute to the place formation process. This approach is comparable with the social constructionist approach, but the construction process is not restricted in social processes but contains a broader range of user activities on the Internet. The resulted place representation usually indicates the distribution of user activities in space and reflects the boundary of the place. L. Li and Goodchild (2012) explore the use of spatial footprints as a record of human interaction with the environment. Specifically, the spatial footprints are geo-tagged photos posted in Flickr. The authors construct a smooth surface according to the density of geo-tagged images in spatial extent. The surface indicates the intensity of spatial points and is interpreted as the significance and location of a place. Goodchild also suggests the use of VGI

or crowdsourcing to formalize the concept of place, highlighting citizen-based initiatives, such as Wikimapia as a new type of gazetteer with the self-proclaimed aim of "describing the whole world" (Goodchild, 2007).

- *The contrastive approach.* The contrastive approach defines a place by comparison. This approach is related to the descriptive approach and the social constructionist approach, where the unique properties of different places are investigated and compared. In the collective place definition, the observations of the same place always have similar properties, while the observations from different places have their distinct properties to distinguish between them. It is necessary to have a relative reference system that allows a comparison of places, such that distinct places can be identified by contrast. Winter and Freksa (2012) approach the notion of place by using the contrast principle in cognition and language. Their contrast principle adopts Tuans perspective about place, i.e., places provide stability and permanence as centers of perceived value, as the basis to construct a relative reference system (Tuan, 1977, p. 29, p. 139).

- *The comprehensive approach.* The comprehensive approach property is a mixture of social constructionist approach and phenomenological approach. In the social constructionist approach, the driven force of place formation in a social process is investigated. However, the comprehensive approach has a broader view of the driving force, such as human perception or emotional response, not only been restricted in the social aspects. Both the cognitive processes and social processes can be chosen as the driven force to form a place. Specifically, in the collective place formation process, the human experience of "in-place" can be obtained from either lab/wild investigation or mined from their digital footprints. The subjective human experience is thus quantitatively connected to the place forming process. The Place Pulse platform[1] collects peoples emotional response to street level images by asking users to choose a safer/livelier/more boring. between two street-level images. Based on the dataset researchers are able to depict the images of different cities from subjective human perceptions, such as safety (Salesses et al., 2013).

To implement the proposed approaches, we derive a probabilistic place representation and use deep learning methods to learn it from data. In detail, CNNs are used to extract essential features of a place from massive image observations. The numeric features are assumed to be generated by a probabilistic process. Thus the visual feature of a place is a random variable of some probabilistic distributions. VAE can be used to learn such a probabilistic place representation, such that the place representation is comprehensive to the observation process. Meanwhile, the learned features and place representations should be put into a relative reference system, so different places can be compared and distinguished. In deep learning, the comparative methods are used to find such a reference

[1]Place Pulse running by MIT Media Lab. http://pulse.media.mit.edu/ last access date: 1.11.2018

system. These methods reflect the contrastive approach by comparing the learned place descriptions.

4.1.2 A Probabilistic Place Model and VAE

A definition of place with latent variable model

Considering an image taken from a place as an observation of the place, we introduce a probabilistic place definition based on the latent variable model. Latent variable model is a widely used probabilistic model that relates an observable variable to a latent variable. In our study, the unobservable latent variable preserves the visual information of a place and thus defines a place, while each image observation captures part of the place's visual information. We denote image observations as $\boldsymbol{x}$ and the latent variable as $\boldsymbol{z}$. We also assume the observations and the latent variables are two continuous random variables. Figure 4.1 is a typical latent variable model in graph representation, where the arrows indicate the interaction between the latent variable and the observation variable. In the next section, we fully explain this figure in the context of VAE. Given the latent representation of a place, we can write its joint distribution with image observation $p(\boldsymbol{x}, \boldsymbol{z})$. To get an image observation of the place, we can use the following generative process given by the latent model,

For each image observation i:

1. Drawing a latent variable $\boldsymbol{z}_i \sim p(\boldsymbol{z})$
2. Drawing an image observation $\boldsymbol{x}_i \sim p(\boldsymbol{x}|\boldsymbol{z}_i)$

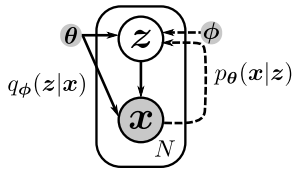


Figure 4.1: The typical latent variable model in directed graph representation, where N represents the number of data samples, and arrows show the dependency relationship between x and z.

Instead of seeking a geometric boundary to define a place, we define a place in a probabilistic way with the latent variable model. The boundary and semantic information of a place are constructively obtained from observations. Knowing the true distribution of the latent model, i.e., the place definition, we can infer all possible observations by a sampling process.

The image observations capture the visual property of a place, even though they may vary in appearance due to different conditions, such as season, weather condition, lighting condition. Tangible objects, such as a building, a tree, or a road sign, all compose the visual property of a place. These objects and their combinations may have the discriminative power to distinguish the place from others and thus connect with the spatial property of a place, such as the positions of these objects or camera poses of image observations. We assume to represent the visual property with a latent variable. The latent variable, on one hand, encodes the visual property of a place, on the other hand, can be distinguished among different places.

Some conditions are irrelevant to specific places, such as lighting conditions, and random noise, while others may be connected with a specific place, such as camera poses. Since we can hardly distinguish these conditions exhaustively, only part of these conditions can be explicitly represented. We refer them as controllable conditions, such as camera pose. Others that are not explicitly considered such as lighting condition are referred to as uncontrollable conditions. For those explicitly contained conditions, we have two different ways to incorporate them with the latent representation, either by using them in the learning process to make them implicitly contained in the latent representation or by explicitly using them as a condition. Since we do not have any supervision information about those uncontrollable conditions, they are inseparable from the latent variable. In the rest of this section, we introduce VAE to facilitate the learning process.

Learning the latent place model with VAE

Suppose each image observation has its unique latent code $\boldsymbol{z}$ that is sampled from the distribution $P(\boldsymbol{Z})$. Generating an image observation of a place equals to sample from the conditional distribution $P(\boldsymbol{X}|\boldsymbol{Z})$. Finding the latent code of an image observation turns to evaluate the conditional distribution $p(\boldsymbol{z}|\boldsymbol{x})$, where the image $\boldsymbol{x}$ is given as the condition. However, it is intractable to integrate over the latent variable for the marginal likelihood $p(\boldsymbol{x}) = \int p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})\mathrm{d}\boldsymbol{z}$. The reason is that the true posterior of the latent variable $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}|\boldsymbol{x})$ is unknown. Kingma and Welling (2013) originally proposed VAE to tackle this issue.

In the context of VAE, the true posterior distribution of image observations $\boldsymbol{X}$ over the latent representation $\boldsymbol{Z}$ can be approximated by a variational distribution $Q(\boldsymbol{Z})$, i.e., $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = q_{\boldsymbol{\phi}}(\boldsymbol{z})$. Suppose there are N observations of a place $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(N)}\}$. All observations are independent identically distributed (the **i.i.d.** condition), and these observations are associated with the latent variable $\boldsymbol{x} \sim p_{\boldsymbol{\theta}^\star}(\boldsymbol{z}|\boldsymbol{x})$, where $\boldsymbol{\theta}^\star$ is the parameter of the true distribution.

VAE approaches the 3 parts of this model, i.e., the parameters $\boldsymbol{\theta}$, the posterior of latent variable $\boldsymbol{z}$ given an input $\boldsymbol{x}$, and the marginal of variable $\boldsymbol{x}$ by introducing an approximation

$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ to the intractable true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. The latent variables $\boldsymbol{z}$ can be seen as an interpretation as a latent representation or code. Thus a VAE model can be divided into an encoder and a decoder. The encoder (or recognizer) is a differentiable parametric function $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, where $\boldsymbol{\phi}$ is the parameter. It approximate the intractable true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ by mapping the image observation to its corresponding latent code in a simpler distribution. The decoder reconstructs an image observation from its latent code, i.e., mapping the latent codes back to the image observations. The decoder function produces a distribution of possible $\boldsymbol{x}$ given a latent code $\boldsymbol{z}$, i.e., it estimates the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. Both encoder and decoder functions can be implemented as neural networks.

The objective of VAE is to maximize the logarithmic marginal likelihood over a dataset as the summation of the logarithmic marginal likelihoods of individual samples, i.e.,

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), \tag{4.1}$$

which can be rewritten as:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) = D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}), \tag{4.2}$$

where the Kullback-Leibler divergence (KL-divergence) D_{KL} measures the difference between the estimated posterior Q with parameters $\boldsymbol{\phi}$ and the true posterior; $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)})$ is called the variational lower bound (also known as the evidence lower bound, ELBO). Since $D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}))$ is non-negative and intractable, we turn to maximize the ELBO. This also leads the recognition model to approximates to the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. The ELBO is written as Equation 4.3. The KL-divergence term in ELBO measures how close the recognized distribution approaches the prior of $\boldsymbol{z}$. The expectation term measures how much information of an input image can be recovered from the latent code. When the recognized distribution is the same as the prior and the information of input images can be perfectly recovered, the ELBO achieves its maximum value. In practice the encoder network estimates the mean and variance of the latent code as a normal distribution to estimate the KL-divergence term.

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = -D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}))\right]. \tag{4.3}$$

The core idea of VAE is to map the unknown prior distribution of the latent variable to a simpler distribution. The idea is shown in Figure 4.2. Suppose we have a set of image observations of a place and a well-trained VAE model. The encoder of the model comprises an image into a latent code while the decoder takes a latent coder and reconstruct the corresponding image observation. Given a simple distribution, e.g., normal distribution as shown in Figure 4.2 (a), we can find corresponding value from the normal distribution for all image observations when they go through the encoder. Similarly, each value in the normal distribution can be mapped back to an image observation using the decoder

network. As shown in Figure 4.2 (b), the VAE actually pushes the unknown complex data distribution towards a standard normal distribution, i.e., the high probability part of the data distribution is squeezed towards the center of the normal distribution (zero value), while the low probability part of the data distribution is dispersed in the opponent directions out of the center. In this case, we establish a connection between the image observations and the standard normal distribution.

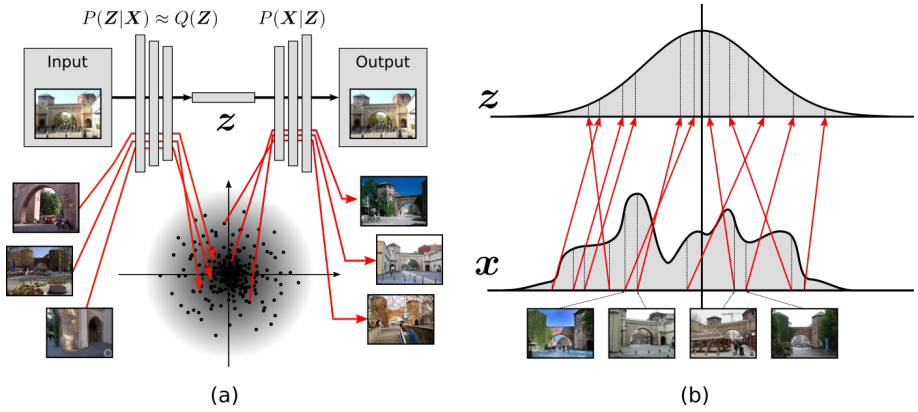


Figure 4.2: The core idea of using VAE in place learning, (a) the encoder maps each image observation to its corresponding latent code while the decoder recovery the image observation from a latent code, (b) VAE maps a complex data distribution to a simple latent distribution.

VAE can be trained with the normal back-propagation algorithm, but the randomly sampled latent variable causes unstable gradient values, so the authors of the original VAE paper propose a reparametrization trick to get stable gradient in training (Kingma and Welling 2013). The reparameterized random variable $\boldsymbol{z}$ is expressed as a "deterministic" form $\tilde{\boldsymbol{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$. For example, if $\boldsymbol{z}$ is a univariate Gaussian, i.e., $z \sim \mathcal{N}(\mu, \sigma^2)$, then it can be reparameterized as $z = \mu + \sigma\epsilon$, where ϵ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$. In this case the ϵ is an input for recognition network and μ, σ are the deterministic output of the network to produce gradient.

4.1.3 Comparative Learning Methods

The motivation of using comparative methods

Though the place recognition problem, which requires to assign observations to their corresponding places and many other application domains, such as face recognition, person re-identification (Re-ID) can be fitted well into a classification setting, comparative methods are more preferable than directly applying classification losses (e.g., softmax cross entropy loss). The motivation of using comparative methods can be summarized as the need for the investigation of learned representations rather than the classification scores.

Moreover, Tadmor et al. (2016) point out that the classification-based training is easier than the comparative training, but the classification-based method requires a large training set because of the significant increase in the number of parameters in the output layer. In such cases, the learned representations are placed in a relative reference system, e.g., a Euclidean space, to facilitate the comparison between different samples. In the place recognition problem, an ideal situation is when images from the same place, their representations are closer while images from different places are positioned far away from each other in terms of the learned representations. In general, the learned representations should have a small intra-class variation and a large inter-class variation. The comparative methods also share the common idea as the contrastive approach for collective place definition. In the rest of this part, we briefly review several comparative methods in deep learning. In the next sections, we show how we combine some of the methods with VAE model for place recognition.

Before we begin to introduce these methods, some concepts and terms are clarified first. Suppose we have a collection of data $\boldsymbol{X}$ (in our case image observations) and a metric distance function in the data space. Let function $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ which takes two data samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ to produce a non-negative scalar value defines the distance between them. We use $\mathcal{D}$ to denote the distance computed by the metric function for short. The function should satisfy the following conditions,

- non-negativity, $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$;
- identity of indiscernible, $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0, \iff \boldsymbol{x}_i = \boldsymbol{x}_j$;
- symmetry, $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_j, \boldsymbol{x}_i)$;
- triangle inequality, $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{x}_k) + d(\boldsymbol{x}_k, \boldsymbol{x}_j)$.

Due to the difficulty of finding such a metric distance function directly in data space (i.e., image space) when the data is high dimensional and complex, we turn to find a function in its latent space. The metric distance function in the latent space enables to compare two data samples with a known distance metric, e.g., Euclidean metric. If we have a parametric function $f_{\boldsymbol{\theta}}$ that takes $\boldsymbol{x}$ as input and produces a vector value as the latent code, we can require the function maps the image data into a latent space so that a metric is applicable in the space. The function is usually implemented as a deep network. Let $\boldsymbol{Z}$ denotes the h-dimension latent variable, $\boldsymbol{z}_i \in \boldsymbol{Z}$ denotes the latent code of data sample $\boldsymbol{x}_i$. The metric distance function in the latent space is denoted as $\mathcal{D}_{\boldsymbol{\theta}} : (\boldsymbol{R}^h, \boldsymbol{R}^h) \mapsto \mathbf{R}_0^+$.

Contrastive Loss

Contrastive loss is originally proposed as the functional component of a dimension reduction method to find meaningful (computable) metric distance in the latent space (Hadsell et al., 2006). The method runs over pairs of samples with binary labels indicating a pair

contains similar of distant data samples. The basic idea of contrastive loss is to construct partial loss for pairs either of two similar samples or of distant samples. The label $y = 1$ if the two samples are similar and $y = 0$ if they are distinct. Given a pair of samples $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and their label y, the loss function can be written as

$$\mathcal{L}(\theta, y, \boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{2}(1-y)(\hat{\mathcal{D}}_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j))^2 + \frac{1}{2}y\{\max(0, m - \hat{\mathcal{D}}_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j))\}^2 \tag{4.4}$$

where $\hat{\mathcal{D}}_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_j))$ and m is the margin value, a hyper-parameter that controls how much difference we want between $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i)$ and $\boldsymbol{x}_j = f_\theta(\boldsymbol{x}_j)$.

The intuition behind contrastive loss is to push latent codes of similar data samples closer and pull latent codes of distinct data samples further than a margin distance.

Triplet Loss

The triplet loss is first introduced in a paper from Google for face recognition (Schroff et al., 2015). It is also used in image retrieval (Wang et al., 2014) and person re-identification (re-ID) (Hermans et al., 2017).

A triplet of latent codes including the following three elements,

- an anchor $\boldsymbol{x}_a$,
- an positive $\boldsymbol{x}_p$ of the same class as the anchor,
- and a negative $\boldsymbol{x}_n$ of a different class,

The core idea of triplet loss is to push the negatives of an anchor point further while pull positives towards the anchor point, until the distance between negatives and positives greater than a margin. The metric function d that measures the distance on the latent space, thus the loss of a triplet $(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n)$ is defined as following,

$$\mathcal{L}(\theta, \boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n) = \max\{\hat{\mathcal{D}}_\theta(\boldsymbol{x}_a, \boldsymbol{x}_p) - \hat{\mathcal{D}}_\theta(\boldsymbol{x}_a, \boldsymbol{x}_n) + m, 0\} \tag{4.5}$$

To minimize this loss, the distances between the latent codes of the same class are pushed towards 0, and the distances between the latent codes of different classes are pulled apart from each other at least at a distance of $\hat{\mathcal{D}}_\theta(\boldsymbol{x}_a, \boldsymbol{x}_p) + m$. As the distance between anchor and negative are greater than the distance, the loss becomes zero.

There are three possible categories of triplets depending on the distances,

- easy triplets, where $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_p) + margin < \hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_n)$, its loss is 0;
- hard triplets, where $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_n) < \hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_p)$, the negative is closer to the anchor than the positive;

- semi-hard triplets, where $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_p) < \hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_n) < \hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_p) + margin$, the negative is farther away from the anchor than the positive but not more than margin.

There are naturally two strategies to make use of these triplets, one is *batch all strategy*, where all valid triplets are selected, and the overall loss is an average over all hard and semi-hard triplets; the other one is *batch hard strategy*, where only triplets with the hardest positives (largest distance among $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_p)$) and the hardest negative (smallest distance among $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}_a, \boldsymbol{x}_n)$) among the batch are selected. Hermans et al. suggest that the batch hard strategy yields the best performance (Hermans et al., 2017).

Center Loss and Its Combination with Comparative Losses

Center loss extends comparative losses by introducing the concept of class centers. A class center is a point in the latent space and approximated by the mean of all latent codes from the data sample of this class. Center loss has been brought up as an auxiliary loss in classification tasks to enhance the discriminative power of deeply learned features (Wen et al., 2016). In the training process, the center loss simultaneously updates class centers and penalizes the distance between the learned features and their corresponding class centers. Center loss is also possible to combine with other comparative losses. X. Zhang et al. (2017) notice the negative effect of long tail distribution in training data and propose a range loss to address this issue. The long tail effect indicates the unbalanced distribution of data in different categories, i.e., the majority of training data comes from a small portion of classes while the majority of the classes have limited number of data samples. The range loss method assumes a center for each class and penalizes on the distance of both intra-class and inter-class cases. Specifically, the loss function consists of two parts respectively, the intra-class loss and the inter-class loss. For each class in a training batch, the method calculates k-greatest range's harmonic mean value of the latent codes for the intra-class loss and uses the margined distance between class centers as the inter class loss. X. He et al. (2018) combine triplet loss and center loss for multi-view 3D object retrieval. This method uses class centers as anchors and calculates the distance between a class center and a positive or a negative sample. It also uses the center loss proposed by Wen et al. as a regularization term to ensure the intra-class similarity (Wen et al., 2016). Center loss involves a set of data $\boldsymbol{x}_k$ with labels indicating their classes and a set of class centers $\boldsymbol{C} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_K\}$. Triplet center loss can be expressed by the following equation, where m indicates a margin value.

$$\mathcal{L}(\theta, \boldsymbol{x}_k, \boldsymbol{C}) = \max\{\hat{\mathcal{D}}_\theta(\boldsymbol{x}_k, \boldsymbol{c}_k) - \frac{1}{K-1} \sum_{i=1, i\neq k}^{K} \hat{\mathcal{D}}_\theta(\boldsymbol{x}_k, \boldsymbol{c}_i) + m, 0\} \tag{4.6}$$

A Brief Summary for Comparative Learning Methods

In Figure 4.3, we illustrate the basic ideas of the three comparative methods. The contrastive loss pushes samples of different classes away and drags the same samples together

as shown in Figure 4.3 (a). After learning, the samples of the same class are close to each other, while the samples of the different classes are at least at the distant of margin value. Figure 4.3 (b) shows the idea of triplet loss, where the anchor is put in the center, the circle is a positive sample, and the triangle represents a negative sample. Triplet loss pulls the positive sample towards the anchor and pushes the negative sample away from the anchor until the distance between the anchor and the positive sample is greater than a margin value compares to the distance between the anchor and the negative sample. In Figure 4.3 (c), a combination of triplet loss and center loss is shown. In this schema, on the one hand, all data samples are pulled towards their corresponding class centers, on the other hand when the distance between a sample and a different class center is less than a margin value, it is also pushed away from the class center. Meanwhile, the class center is also adjusted according to the positions of its corresponding data samples.

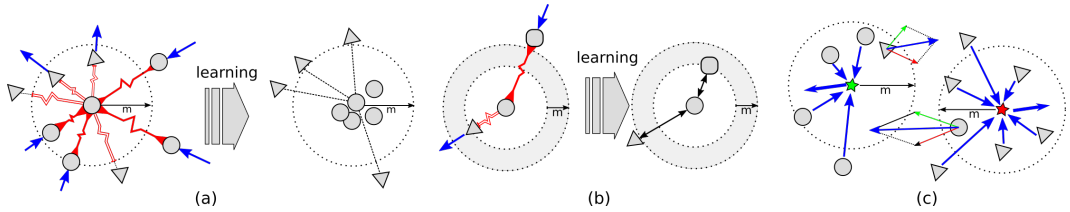


Figure 4.3: A summary of comparative methods in deep learning, (a) contrastive loss, (b) triplet loss, (c) triplet center loss.

4.2 The Combination of Comparative Methods and VAE for Place Representation Learning

In this section we combine triplet loss methods with VAE, so that a mapping from the image space to the latent space can be found to facilitate the comparison in the latent space.

4.2.1 Triplet-VAE for Place Representation Learning

We assume the latent place representation is a random variable of a simple distribution. Generating an image observation is turned into a sampling process, which consists of two steps, 1). sample a latent code $\boldsymbol{z}$ from its prior distribution $P(\mathbf{Z})$, and 2). sample the image observation via the posterior $p(\boldsymbol{x}|\boldsymbol{z})$. Given the labeling information that the observation image data comes from several different place, we can use the Triplet-VAE model to refect places in the latent space. The Triplet-VAE model combines triplet loss and VAE to learn representations of different places.

In the Triplet-VAE model, the triplet formation is seen as a probabilistic process, where the latent codes are sampled from the latent variable distribution and a Bernoulli likeli-

hood is used to indicate whether randomly sampled three elements form a valid triplet (Karaletsos et al., 2015). Given the supervision information, we can verify whether three randomly drawn samples from the latent space form a valid triplet. Let $valid(\cdot)$ denote a the verification function which takes three samples $(i, j, k), i \neq j \neq k$, and produces a real scalar v in $0, 1$, where 1 stands for a valid triplet group and 0 vice versa. For any group of $(\boldsymbol{z}^{(i)}, \boldsymbol{z}^{(j)}, \boldsymbol{z}^{(k)})$ we can denote its probability distribution of being a valid triplet as $p(valid(i, j, k) = 1)$, if (i, j, k) forms a valid triplet then $p(valid(i, j, k)) = 1$, other wise $p(valid(i, j, k)) = 0$. The distribution of sampling valid triplets is a Bernoulli distribution. The joint distribution of N data samples and L potential triplet groups can be written as (Equation 4.7),

$$p(\boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta}) = \int_{\boldsymbol{z} \in \mathbf{Z}} \prod_{n}^{N} \left[p(\boldsymbol{z}^{(n)}) p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(n)} | \boldsymbol{z}^{(n)}) \right] \prod_{l}^{L} \left[p(v^{(l)} | \boldsymbol{z}^{(l_i)}, \boldsymbol{z}^{(l_j)}, \boldsymbol{z}^{(l_k)}) \right] \mathrm{d}\boldsymbol{z}. \tag{4.7}$$

The probability of a triplet group of latent codes is a valid one can be estimated by the following,

$$p(v^{(l)} | \boldsymbol{z}^{(l_i)}, \boldsymbol{z}^{(l_j)}, \boldsymbol{z}^{(l_k)}) = \frac{e^{-\hat{\mathcal{D}}_{i,j}}}{e^{-\hat{\mathcal{D}}_{i,j}} + e^{-\hat{\mathcal{D}}_{i,k}}}. \tag{4.8}$$

The joint distribution can be rewritten as Equation 4.9. The corresponding variational lower bound with an extra term of triplet sampling is given as Equation 4.10,

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{v}) = D_{\mathrm{KL}}(q(\boldsymbol{z}) || p(\boldsymbol{z} | \boldsymbol{x}, \boldsymbol{v})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{v}), \tag{4.9}$$

where

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{v}) = & - \mathbb{E}_n \left[D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}^{(n)} | \boldsymbol{x}^{(n)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z})) \right] \\ & + \mathbb{E}_n \left[\mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(n)} | \boldsymbol{z}^{(n)}) \right] \\ & + \mathbb{E}_k \left[\mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}, v)} \log p_{\boldsymbol{\theta}}(v^{(l)} | \boldsymbol{z}^{(l_i)}, \boldsymbol{z}^{(l_j)}, \boldsymbol{z}^{(l_k)}) \right]. \end{aligned} \tag{4.10}$$

To maximize the likelihood of the above joint distribution we want to minimize the KL-divergence term and maximize the other two expectations. The process of finding an optimization to maximize the likelihood can be fitted into the original VAE training process. We only focus on the last term, the expectation of v given $\boldsymbol{z}$. If we always draw samples from the valid triplets, the probability distribution $p(valid(i, j, k) = 1)$ should always be 1, which means the expectation should be maximized towards 1. In practice, to maximize the triplet exception, we can just minimize the triplet loss to make the triplets as valid as possible.

KL-divergence measures how one distribution is different from another one, but not conversely, i.e., $D_{\mathrm{KL}}(P||Q) \neq D_{\mathrm{KL}}(Q||P)$ (P and Q are two probability distributions). Hence, KL-divergence is not a distance metric. Jensen-Shannon divergence(JS-divergence) is a metric that measures the similarity between two probability distributions. JS-divergence

is symmetry and always a finite value compare to KL-divergence. For two probability distributions P and Q their JS-divergence is defined as the following,

$$D_{\mathrm{JS}}(P||Q) = \frac{1}{2}D_{\mathrm{KL}}(P||M) + \frac{1}{2}D_{\mathrm{KL}}(Q||M), \tag{4.11}$$

where $M = \frac{1}{2}(P + Q)$. Thus the metric function in the latent space can be defined with the JS-divergence as

$$\hat{\mathcal{D}}_{\boldsymbol{\theta}} = D_{\mathrm{JS}}(q_{\boldsymbol{\phi}}(\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{x})). \tag{4.12}$$

However the JS-divergence is commonly intractable due to the difficulty of knowing the intermediate distribution M, and we use an approximation as Equation 4.13 instead.

$$D(P||Q) = \frac{1}{2}D_{\mathrm{KL}}(P||Q) + \frac{1}{2}D_{\mathrm{KL}}(Q||P). \tag{4.13}$$

If P and Q are two normal distributions, i.e., $\mathcal{N}_1(\mu_1, \sigma_1^2)$ and $\mathcal{N}_2(\mu_2, \sigma_2^2)$, the KL-divergence has a closed-form as Equation 4.14,

$$D_{\mathrm{KL}}(\mathcal{N}_1||\mathcal{N}_2) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2}. \tag{4.14}$$

The alternative distance metric is

$$D_{\mathrm{KL}}(\mathcal{N}_1||\mathcal{N}_2) = \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} + \frac{\sigma_2^2 + (\mu_2 - \mu_1)^2}{2\sigma_1^2} - 1. \tag{4.15}$$

When $\mu_1 = \mu_2$ and $\sigma_1 = \sigma_2$, the distance gives 0, otherwise, it produces a positive real value.

In Chapter 5, we implement this method for the first experiment. We describe the network structure, dataset, as well as the results and discussions in the corresponding section.

4.2.2 Learning Gaussian Mixture Place Representations (GMPR)

In the Triplet-VAE model, the prior distribution of the latent place representation is assumed as a simple distribution such as the standard Gaussian distribution. The information about different places is implicitly encoded in the latent codes via triplet loss. In this section, we explicitly encode place information as unique distributions via mixture models. This leads to Gaussian Mixture Place Representations, where multiple places can be represented as unique Gaussian distributions.

Representation of Places via Mixture of Gaussians

Gaussian mixture model is a linear combination of more than one basic Gaussian distributions for better approximation to the data distribution. By adopting the concepts of Gaussian Mixture Model, we now formally describe a Gaussian Mixture Place Representation. Given a collection of N observations $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(N)}\}$ that are captured from K different places $C = \{c_1, c_2, \cdots, c_K\}$, we assume their latent codes $\boldsymbol{Z} = \{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(n)}\}$ are generated from their corresponding probabilistic place representations. Now introduce a K-dimensional binary variable $\boldsymbol{c}$ indicates from which place an observation comes. When the latent representation $\boldsymbol{z}_{c_k}$ (corresponding to the image observation $\boldsymbol{x}_{c_k}$) is related to place c_k, the k^{th} dimension of $\boldsymbol{c}$, i.e., c_k is set to 1 and the rest are set to 0. The probability distribution of a latent code given place indicator $\boldsymbol{c}$ can be evaluated by the conditional distribution $p(\boldsymbol{z}|\boldsymbol{c})$ as

$$p(\boldsymbol{z}|\boldsymbol{c}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{z}|\mu_k, \Sigma_k)^{c_k}. \tag{4.16}$$

Let $\boldsymbol{\pi}$ denotes the distribution probability of place indicator variable and $\pi_k = p(c_k = 1)$, the distribution of the latent code $\boldsymbol{z}$ over K places can be obtained by marginalizing out the place indicator $\boldsymbol{c}$ as,

$$p(\boldsymbol{z}) = \sum_{k=1}^{K} p(\boldsymbol{z}|\boldsymbol{c})p(\boldsymbol{c}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4.17}$$

Given a latent code $\boldsymbol{z}$, the probability of which place belongs to can be estimated by the conditional distribution using Bayes' theorem as,

$$p(c_k = 1|\boldsymbol{z}) = \frac{\pi_k \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{4.18}$$

Equation 4.18 is also viewed as the responsibility that component k takes for "explaining" the latent code $\boldsymbol{z}$.

Back to the image observations, we are interested the relationship between an image observation and its latent code. Since usually we have no clue about the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$ which "encodes" the image observations to their latent codes, we can use a variational distribution instead as VAE. We are also interested in the posterior distribution $p(\boldsymbol{x}|\boldsymbol{z})$ which "decodes" a given latent code to its corresponding image. Generating a random image observation is expressed by the following process, where $p(\boldsymbol{c})$ is the prior distribution of place indicators and $p(\boldsymbol{z}|\boldsymbol{c})$ is the distribution of the corresponding latent code given $\boldsymbol{c}$, where $\boldsymbol{\psi}$ is used to denote the parameters of Gaussian components.

1. Choosing a place indicator, $\boldsymbol{c} \sim \text{Cat}(\boldsymbol{\pi})$,
2. Sampling a latent code from place representation, $\boldsymbol{z} \sim P(\boldsymbol{z}|\boldsymbol{c})$,
3. Sampling an image observation $\boldsymbol{x}$, $\boldsymbol{x} \sim P(\boldsymbol{x}|\boldsymbol{z})$

Center-Triplet-VAE for GMPR Learning

A functional GMPR model consists of three parts, the "encoder" that encodes an input image into its latent code, the "decoder" recovers an image from the latent code and the place representation as to the parameters of Gaussian components. We propose the Center-Triplet-VAE model to learn such a functional GMPR model. Center-Triplet-VAE (CTV) is inspired by the center loss (Wen et al., 2016) and Center-Triplet method (X. He et al., 2018). The point of the mean value of a Gaussian component defines a "class center" in the latent space, while the latent codes of image observations that come from the same place are considered as positives while the latent codes from a different place are considered as negatives. Instead of using a Euclidean distance in the center loss, we use probability to indicate how close a latent code to a place representation. As the probability cannot define a metric function, it actually measures the distances from a given latent code to all class centers. The idea of CTV is shown in Figure 4.4.

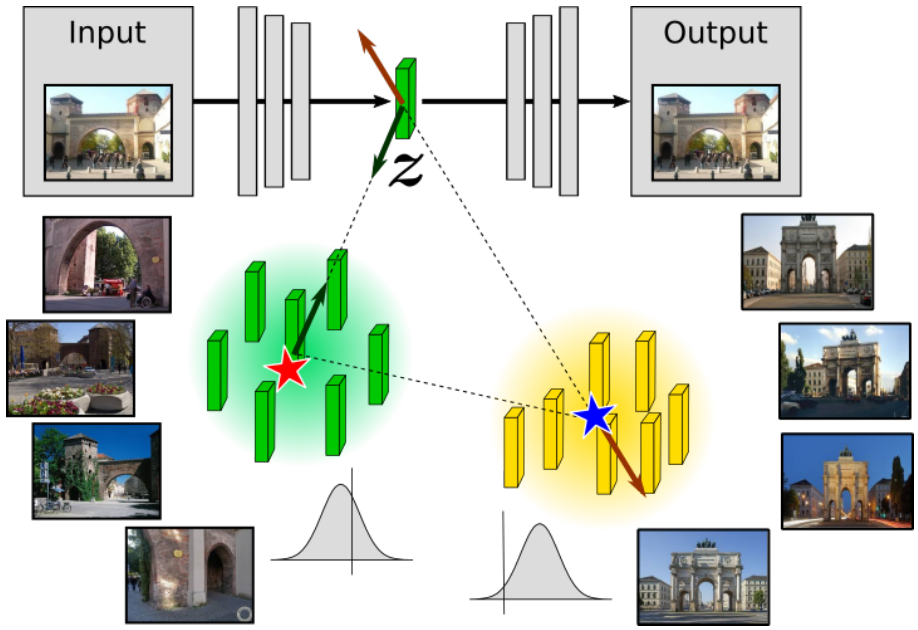


Figure 4.4: The schematic drawing of CTV model, where two different places and their representations are shown. The loss of CTV, on the one hand, pushes the latent codes from different places away, as well as their corresponding representations, on the other hand, ensures the latent codes of the same place as close to their corresponding distribution as possible.

The Center-Triplet-VAE (CTV) model on a collection of images from different places is trained to maximize the likelihood of the given N observations and place labels. The log-likelihood of CTV $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}, \boldsymbol{c}^{(1)}, \cdots, \boldsymbol{c}^{(N)}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})$ can be

written as

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) = D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}). \tag{4.19}$$

The first right hand side (RHS) term is the KL divergence between the approximated posterior distribution where only $\boldsymbol{x}^{(i)}$ is given as a condition and the true posterior where both $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(i)}$ are given. The second right hand side term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})$ is the variational lower bound (ELBO) on the marginal likelihood of image observation i as,

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) = -D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \log p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}|\boldsymbol{z}). \tag{4.20}$$

$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ is the variational posterior to approximate the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})$. For the GMPR model, we assume $p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}|\boldsymbol{z})$ can be factorized as:

$$p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}|\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) p_{\boldsymbol{\psi}}(\boldsymbol{c}^{(i)}|\boldsymbol{z}). \tag{4.21}$$

Then, the ELBO in Equation 4.20 can be rewritten as:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) = & - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})) \\ & + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \\ & + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \log p_{\boldsymbol{\psi}}(\boldsymbol{c}^{(i)}|\boldsymbol{z}). \end{aligned} \tag{4.22}$$

We want to differentiate and optimize the lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})$ w.r.t. the variational parameters $\boldsymbol{\phi}$, the generative parameters $\boldsymbol{\theta}$, and the parameters of the Gaussian components $\boldsymbol{\psi}$. The gradient of the lower bound can be computed by the Stochastic Gradient Variational Bayes (SGVB) proposed in the original VAE paper with a "reparameterization trick" (Kingma & Welling, 2013).

Now we provide some intuitions of the ELBO as Equation 4.22. More specifically, the ELBO of the CTV adds a strong assumption about the distribution of $\boldsymbol{z}$ according to the GMPR configuration. For each pair of observation samples $< \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)} >$, the encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ approximates the Gaussian component that is indicated by $\boldsymbol{c}^{(i)}$, i.e., $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})$. The assumption to factorize the joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}|\boldsymbol{z})$ suggests that both the image observation $\boldsymbol{x}^{(i)}$ and its latent code $\boldsymbol{z}^{(i)}$ contain the place information $\boldsymbol{c}^{(i)}$. The second RHS term in Equation 4.20 is thus rewritten as the last two RHS terms in Equation 4.22. $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z})$ ensures the visual information in an input image can be restored from its corresponding latent code as the same in the original VAE (see the second RHS term in Equation 4.3). The term $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \log p_{\boldsymbol{\psi}}(\boldsymbol{c}^{(i)}|\boldsymbol{z})$ guarantees that the latent code preserves the place information as well. In CTV, $p_{\boldsymbol{\theta}}(\boldsymbol{c}^{(i)}|\boldsymbol{z})$ is evaluated by Equation 4.18. The mean value of each Gaussian component can be seen as "class center". The distance between a latent code and a class center is thus evaluated as a probability value. In practice, we introduce the center triplet loss similar to Equation 4.6 requiring the probability of a latent code belongs to its class at least greater than a certain value (between 0 and 1) comparing to the probabilities of belonging to other classes. This term also prevents the Gaussian parameters of different components collapsing into the same value.

Weakly Supervised Place Learning

The Center-Triplet-VAE model is developed to learn GMPR when places are pre-defined. We may notice that given an image observation and its class label the posterior is a Gaussian distribution. The distribution of all latent codes over all places is thus a Gaussian Mixture model. When there are no pre-defined place classes, the classes of image observations need to be evaluated as probabilities over all possible places. In this section, we first use a GMM framework to tackle this problem and then introduce geographic constraints as a weak supervision signal to facilitate the learning process.

When no pre-defined place information is given, the model is expected to some extent as a kind of unsupervised generative approach to perform clustering. The generative process is similar as we described in the previous section. Specifically, suppose there are K places (clusters), an observed image is generated by the following process,

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{c}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c}), \tag{4.23}$$

and

1. Sampling a place indicator from its prior, $\boldsymbol{c} \sim P(\boldsymbol{c})$
2. Sampling a latent code from the corresponding place representation, $\boldsymbol{z} \sim P(\boldsymbol{z}|\boldsymbol{c})$
3. Sampling an image observation $\boldsymbol{x}$, $\boldsymbol{x} \sim P(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})$

While the place label is missing, we want to maximize the probability over all image observations with an unobserved label variable $\boldsymbol{c}$ and a latent place representation $\boldsymbol{z}$ that is related to $\boldsymbol{c}$. The log likelihood of $\boldsymbol{x}$ is $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log \int_{\boldsymbol{z}} \sum_{\boldsymbol{c}} p_{\boldsymbol{\theta}}(\boldsymbol{c}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c}) \mathrm{d}\boldsymbol{z}$. Note that we will omit the index i whenever it is clear that we are referring to terms associated with a single observation. If we assume $p_{\boldsymbol{\theta}}(\boldsymbol{c}, \boldsymbol{z}|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$, the variational lower bound becomes,

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x})\right) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) \tag{4.24}$$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x})}\left[\log(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x})} + \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})} + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right]. \tag{4.25}$$

The first term $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x})}\right]$ seems to be anti-clustering, while avoid the place information leak from the latent space to observation space. The second term implies that the latent code acts as intermediary between the place information and the image observation. The last term uses both the latent code $\boldsymbol{z}$ and place information to reconstruct the image observation. We can further rewrite the new variational lower bound as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x})} \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{c})} - \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c})} + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c}) \tag{4.26}$$

Though this model performs clustering on the observed image dataset from a pure data perspective, it misses the spatial property between places and their observations. We assume that the geographically closed images are more likely from the same place while two images that geographically far away from each other are taken from two different places. This assumption motives us to add geographical constraints to the learning process. We add a contrastive loss term to the variational lower bound to reflect the additional geographical constraints. The contrastive loss involves a seed data sample and two groups, the positive group, and the negative group. Data samples in the positive group are expected from the same place as the seed sample while data samples in the negative group are from different places. In our probabilistic model, we introduce Gaussian components to define class centers. Instead of measuring Euclidean distance between two image observations in the latent space (i.e., measuring the distance between their latent codes), we use the probabilistic value $p(\boldsymbol{c}|\boldsymbol{z})$ to measure how much an image belongs to a place in the latent space. We write out the value of $p_{\boldsymbol{\theta}}(\boldsymbol{c}|\boldsymbol{z})$ as $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{c})$. Given a seed image $\boldsymbol{x}_s$ and its location, a set of N images within a certain distance D_+ of $\boldsymbol{x}_s$ is selected as the target group and a set of M images further than a certain distance D_- is selected as the contrastive group. We use $\hat{\mathcal{D}}_{\boldsymbol{\theta}}(\boldsymbol{x}, c_k)$ to denote the probability of $\boldsymbol{x}$ belonging to the place representation that indexed by k, i.e., the component of the k^{th} Gaussian distribution. Since the two groups are geographically far away, they are highly possible from different places, such that their activations should be different. Thus, we set the contrastive loss of the i^{th} contrastive group with marginal value m as,

$$\mathcal{L}_i = \sum_{c=1}^{|C|} (\max(||\hat{\mathcal{D}}(f_+, c), 0) - \hat{\mathcal{D}(f_-}, c)|| - m, 0). \tag{4.27}$$

By combining the ELBO and this contrastive loss, we can learn a GMM-based representation for several places. In the second part of the second experiment in Chapter 5, we proof its feasibility of learning place representation without much prior knowledge of a place.

4.3 Holistic Place Representation with Camera Pose

In this section, we address two issues related to latent place representation. The first issue is how can we disentangle the uncontrollable conditions with the latent place representation. The second issue is how to incorporate camera pose in the latent place representation learning explicitly.

4.3.1 The Holistic Latent Place Representation

In the latent holistic place representation, we distinguish between three components from image observations. They are the latent place representation, the variable for controllable condition, i.e., camera pose in this work, and a latent variable for the uncontrollable

conditions. The latent place representation is a latent variable $\boldsymbol{r}$, and a latent variable $\boldsymbol{z}$ is used to denote the uncontrollable conditions in image observations. Both latent variables are treated as continuous normally distributed variables. The controllable condition in our work is camera pose that is attached to each image observations.

Suppose we have a collection of images taken from some places. The places are indexed by n. The set of images taken from place n is denoted as $\mathcal{X}_n$, which is consisted of its an image observations $\boldsymbol{x}_n^{(m)}$. Each image observation has a camera pose or view attached. The view is denoted as $\boldsymbol{v}_n^{(m)}$. Thus we have pairs of such image observation, and its camera pose to depict a place. We can also assume that, the uncontrollable condition, and the holistic place representation are independent from each other, i.e., $p(\boldsymbol{z}, \boldsymbol{r}) = p(\boldsymbol{z})p(\boldsymbol{r})$, while an image observation depends on both latent variables and its camera pose. The problem can be formulated from the following settings,

$$\begin{aligned}
\text{Observation Images} \quad & \mathcal{X}_n = \left\{\boldsymbol{x}_n^{(1)}, \cdots, \boldsymbol{x}^{(m)_n}\right\} \\
\text{Corresponding Camera Pose} \quad & \mathcal{V}_n = \left\{\boldsymbol{v}_n^{(1)}, \cdots, \boldsymbol{v}_n^{(m)}\right\} \\
\text{Place Representation} \quad & \boldsymbol{r}_n \sim P(\boldsymbol{R}) \\
\text{Uncontrollable Condition} \quad & \boldsymbol{z}_n P(\boldsymbol{Z}),
\end{aligned}$$

In this latent place representation model, we extend the previous model by disentangling the uncontrollable condition and the place representation. The problem setting is somewhat similar to Generative Query Network (GQN) (Eslami et al., 2018), while the difference is that we use a latent variable for place representation instead of a deterministic one. In this research, the authors propose a VAE-based system, GQN to generate view dependent images in a virtual environment. The authors separate the latent representation and camera poses, and use camera pose as a condition in their generative framework. They define an accumulative place representation that consists of several additive components. The components are dependent on the view specified by an image. The authors employ a neural network to produce such view dependent components. The place representation is thus accumulated from multiple components. The resulted place representation is supposed to be a holistic one that contains all necessary information of a place and is independent from views. In this section we propose another latent place representation model similar to GQN, but with a different place representation schema. This representation separates the controllable and uncontrollable conditions to explicitly use camera pose in the VAE model as supervision information.

4.3.2 VAE for Holistic Place Representation Learning

The holistic latent place representation enables us to generate an image observation given the place representation, and a camera pose. The latent variable for the uncontrollable conditions which we can hardly model adds some randomness to the generated image observation. To describe how we can get the uncontrollable variable and the place repre-

sentation, we begin with the generation of an image observation given the uncontrollable condition $\boldsymbol{z}$, place representation $\boldsymbol{r}$, and camera pose $\boldsymbol{v}$,

For an image observation i,

1. Drawing from the uncontrollable condition $\boldsymbol{z}_i$,
2. Drawing from the place representation $\boldsymbol{r}_i$,
3. Drawing an image observation given the uncontrollable condition, place representation, and a camera pose $\boldsymbol{x}_i \sim p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{v})$.

We need to maximize the likelihood of the given data points (observed images),

$$\log \iint_{\boldsymbol{z},\boldsymbol{r}} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{r}|\boldsymbol{v}) \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{r}.$$

Given the assumption on $p(\boldsymbol{z})$ and $p(\boldsymbol{r})$, and the generation process, we can maximize the following log-likelihood with Jensen's inequality,

$$\begin{aligned} \log p(\boldsymbol{x}|\boldsymbol{v}) &= \log \iint_{\boldsymbol{z},\boldsymbol{r}} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{r}|\boldsymbol{v}) \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{r} \\ &\geq \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{r}|\boldsymbol{x},\boldsymbol{v})}\Big[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{r}|\boldsymbol{v})}{q(\boldsymbol{z}, \boldsymbol{r}|\boldsymbol{x}, \boldsymbol{r})}\Big] = \mathcal{L}_{\mathrm{ELBO}}, \end{aligned} \tag{4.28}$$

where $\mathcal{L}_{\mathrm{ELBO}}$ is the variational lower bound (i.e., evidence lower bound, ELBO), $q(\boldsymbol{z}, \boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v})$ is the variational posterior to approximate the true posterior $p(\boldsymbol{z}, \boldsymbol{r}|\boldsymbol{x}), \boldsymbol{v})$. The ELBO can be further rewritten as

$$\begin{aligned} \mathcal{L}_{\mathrm{ELBO}} &= \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{r}|\boldsymbol{x},\boldsymbol{v})}\big[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{v}) + \log p(\boldsymbol{z}) + \log p(\boldsymbol{r}) && (4.29) \\ &\quad - \log q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v}) - \log q(\boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v})\big] && (4.30) \\ \mathcal{L}_{\mathrm{ELBO}} &= \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{r}|\boldsymbol{x},\boldsymbol{v})}\big[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{v})\big] - D_{\mathrm{KL}}(q(\boldsymbol{z}, \boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v})||p(\boldsymbol{z}, \boldsymbol{r})) && (4.31) \\ &= \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{r}|\boldsymbol{x},\boldsymbol{v})}\big[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{v})\big] && (4.32) \\ &\quad - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v})||p(\boldsymbol{z})) - D_{\mathrm{KL}}(q(\boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v})||p(\boldsymbol{r})) && (4.33) \end{aligned}$$

From Equation 4.33, we can see the ELBO to be maximized is composed of three terms, the first term represents the reconstruction of an image observation given all necessary conditions, the second and third term measure the differences between the learned distribution of the uncontrollable conditions and the place representation and their prior respectively. Since two KL divergence terms are always greater than zero, to maximize this ELBO, we need to on one hand maximize the reconstruction term, and on the other hand minimize the two KL divergence terms. Only when the two KL divergence terms are zero, the optimal can be achieved.

To implement this model, we use a recognizer function $q_{\boldsymbol{\theta}}(\cdot)$ to approximate the true posterior $p(\boldsymbol{z}, \boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v})$, and a decoder function $g_{\boldsymbol{\phi}}(\cdot)$ for the conditional distribution. The model can summarized as the following, where both functions can be implemented by parametric deep networks.

$$\begin{aligned} \text{Recognizer} \quad & q_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v}) \\ \text{Decoder} \quad & g_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{v}) \end{aligned}$$

$$\begin{aligned} \text{Prior Uncontrollable Variable} \quad & p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \text{Prior Latent Place Representation} \quad & p_{\boldsymbol{\theta}}(\boldsymbol{r}|\boldsymbol{x}, \boldsymbol{v}) = p_{\boldsymbol{\theta}}(\boldsymbol{r}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{r}}, \sigma_{\boldsymbol{r}}^2 \mathbf{I}) \end{aligned}$$

Unlike the uncontrollable condition, we don't assume a known prior distribution for the place representation. The place representation is learned during the training process. The model can be trained without much difference to the original VAE. In Chapter 5, we illustrate the feasibility of this place representation with the third experiment.

Chapter 5

Experiments

In this section, we present three experiments on place representation learning and image observation sampling. In the first experiment, we use Triplet-VAE to learn place representations and identify latent code clusters of pre-defined places. We also investigate the model's ability to decode a latent code and sample novel images, which suggest the learned latent code preserves the visual information of a place. The second experiment consists of two sub-experiments on the GMPR. In the first sub-experiment, we use Center-Triplet-VAE model to learn such place representation given pre-defined place information and investigate the discriminative power of the learned latent codes. The other sub-experiment targets on a more complex situation without pre-defined place information. In the third experiment, we show how to embed learned place description for view-dependent image generation.

5.1 Learning Places Representations with Triplet-VAE

In this experiment, we illustrate the feasibility of using Triplet-VAE to learn place representation given pre-defined place categories. To evaluate its effectiveness, we set up two baseline models beside Triplet-VAE model, one is an autoencoder with triplet loss, and the other is purely a variational autoencoder. All three models use the same encoder and decoder network structure and the same hyper-parameters. We empirically compare the three models in the following aspects,

- Does the latent place representation learned by Triplet-VAE can be used to distinguish different places?
- To what extent the latent representation captures the visual property of a place?

For the first question, we examine whether the image observations in latent space are clustered according to their place information and whether the latent space generalizes well on the test data samples. To answer the second question, we compare their ability to generate new image sample for different places.

5.1.1 The Model

We adopt similar network architectures as the encoder-decoder structure for the three models. The VAE structure consists of an encoder network and a decoder network, producing the value of latent variable and reconstructed input given an input respectively. We assume the prior $p(z)$ is a standard normal distribution i.e., $\mathcal{N}(0, 1)$, and the encoded latent variable is a normal distribution as well. The probability distribution of possible values for latent variable is described by mean and variance. Thus the encoder needs to predict the mean and deviation at the same time. The latent code is sampled from the normal distribution with the predicted mean and variance. In training phase, we also use the reparameterization trick as Equation 5.1, which, as (Kingma & Welling, 2013) suggests, to obtain stable gradients.

$$\boldsymbol{z} = \mu(\boldsymbol{x}) + \sigma(\boldsymbol{x}) * \epsilon, \qquad \text{where}\, \epsilon\, \mathcal{N}(0, 1). \tag{5.1}$$

In the encoder network, we use convolutional layers (Conv layers) to extract features from input images, and in decoder network, transpose convolution layers (TransConv layers) are used to reconstruct images from the latent code. We also use batch normalization after the Conv/TransConv layers, and Leaky ReLU/ReLU as the activation function. Batch normalization transforms its input signal into a standard normal distribution to solve the internal covariate shift problem (Ioffe & Szegedy, 2015). In practice, using batch normalization can accelerate training and improve models' performance. We refer to the composition of a Conv/TransConv layer, batch normalization layer, and an activation layer as a Conv/TransConv block. Figure 5.1 depicts the overall network structure. The encoder consists of 5 Conv blocks followed by a fully connected layer to make predictions. The decoder consists of a fully connected layer and 4 TransConv blocks with the last block using a Tanh activation. The latent code is first transformed by the fully connected layer and then reshaped for the following transpose Conv blocks. We set strides in both spatial extensions as 2 to replace the pooling layer to reduce the spatial extent of input image and feature maps. The strides 2 in decoder network helps to scale up the generated images. Given the structure of this network, it handles input images of size (64, 64). The settings of each layer, including kernel size and the number of output channel, is detailed in Figure 5.1 as well.

5.1.2 Data

We test our method on the Microsoft 7-scenes dataset (Glocker et al., 2013). The 7-Scenes dataset captures color and 3D information of 7 discrete indoor scenes. The indoor scenes are 7 places including a room with a chess board and two monitors (chess); a set of two fire extinguisher at a room corner(fire); an office room with computer and heads sculptures (heads); an office room with table, computer, and a shelf; a kitchen with red furniture and a pumpkin on the floor (pumpkin); the red kitchen without the pumpkin; and a staircase (stair). The 7-Scenes dataset is collected by a handheld Kinect RGB-D

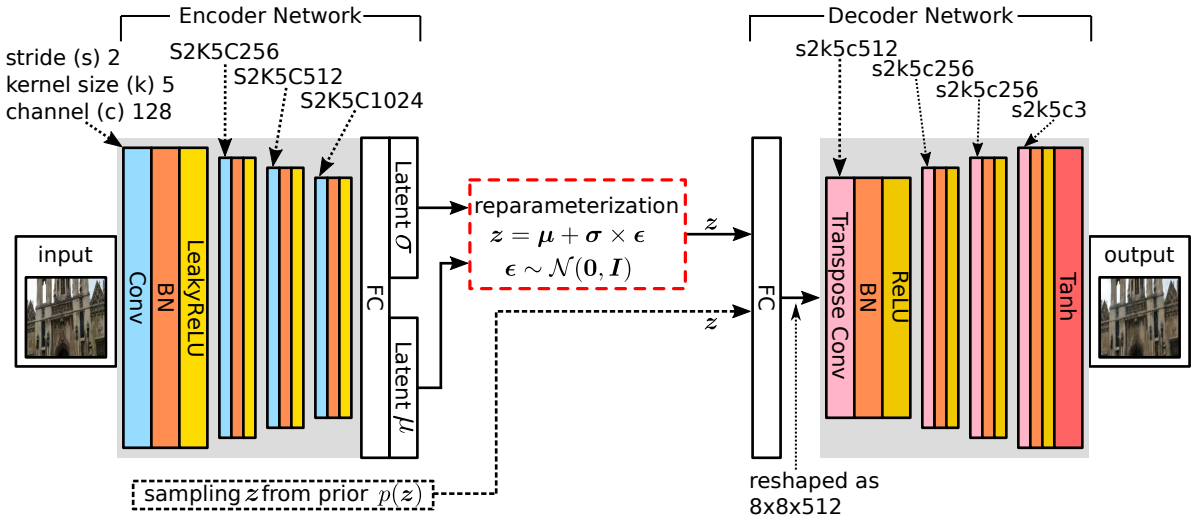


Figure 5.1: The VAE architecture with Conv layers in encoder network and TransConv layers in decoder network.

camera at 640 * 480 resolution. The dataset contains RGB images, depth images, and camera poses. We use RGB images to provide visual features and the scenario names as their class labels. The dataset is originally divided into two parts, one for training and the other for testing. Since all images are captured by the same device, we do not need to consider the influence of the intrinsic parameters of the camera, and thus enable us to focus on the visual feature extraction. Figure 5.2 provides an overview of this dataset, including a sample image, training set trajectories (red), and test set trajectories (green).

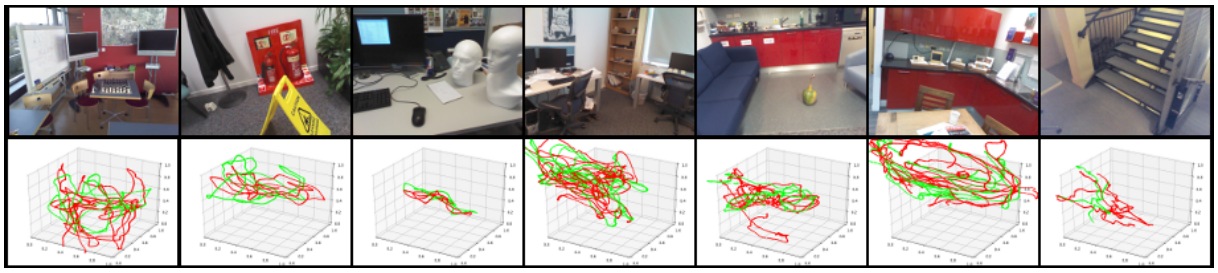


Figure 5.2: An overview of 7-scenes dataset, in the first row a sample image of each scene is given, in the second row, camera poses of training images (in red) and test images (in green) are plotted.

5.1.3 Experiment Settings

In our experiment, we randomly select 500 images from each scene to form a training set and 20 images for testing purpose. In total, there are 3500 images for training and 140 images for testing. The original partitions for training and testing are not affected in our case as long as we assume the images in both datasets are generated from the same stochastic process. The numbers of images for each scene are various, but we select the same number of images for each scene to avoid the potential influence of class imbalance. The imbalanced number of training images for different classes may introduce bias in training. Generally speaking, a scene that has more training images gets more "attention" by the training algorithm while a scene with less training images may be ignored to some extent. To balance the value of different terms in the loss function, we add two composing parameters λ_{KL} and λ_{triplet}, which are pre-defined before training. The loss function is thus represented as

$$\mathcal{L} = \mathcal{L}_{\text{reconstruction}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL-divergence}} + \lambda_{\text{triplet}}\mathcal{L}_{\text{triplet}} \tag{5.2}$$

The hyper-parameters and training details are listed in Table 5.1.

Table 5.1: Training Parameters

lambda(KL)	lambda(triplet)	margin	Learning rate	Batch size	Epochs	Weight Init
0.05	0.05	5.0	0.001	128	20000	$N(0, 0.02)$

All three models are trained on a computer with the following settings. Training each model takes about 40 hours.

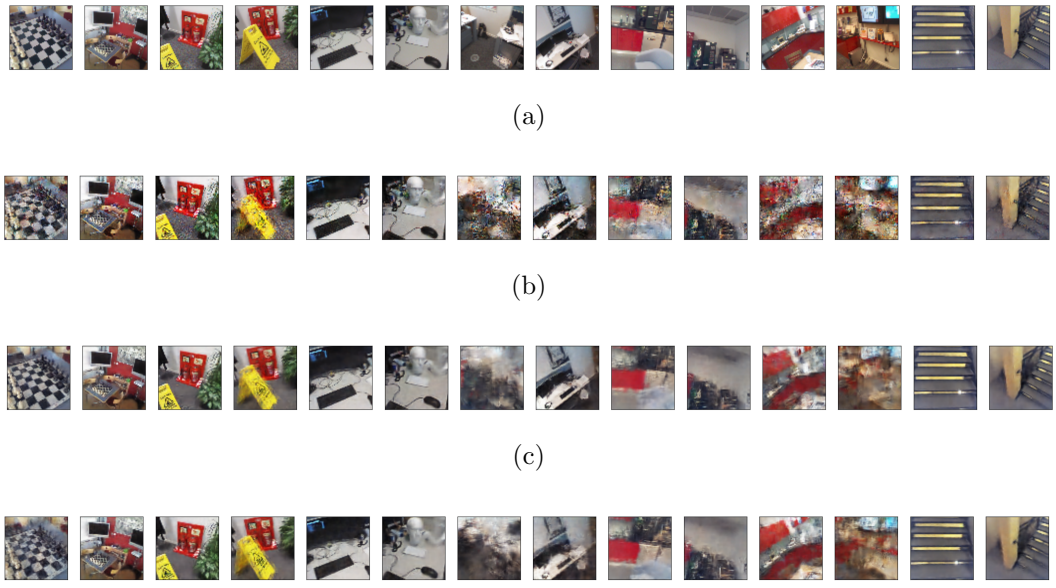
Table 5.2: Environment Details

CPU	Intel Xeon(R)CPU W3580 @ 3.33GHz x 8
Memory	16GB
GPU	Nvidia TITAN Xp GPU
OS	Ubuntu 16.04.5 LTS (x64)
	TensorFlow 1.10.0
	Python 3.5

5.1.4 Results

We first show the encoding-decoding ability of the three models. The encoding-decoding ability is to map an input image into latent space and recovers the image from the latent code. In Figure 5.3, we show the results of the three models based on some randomly selected test images. The first row shows the original images in the dataset with two images for each place (5.3(a)). The rest rows illustrate the decoded images by autoencoder, VAE,

and VAE with triplet loss accordingly. The complete list of all 140 test images and their reconstructions are attached in the **Appendix D**. Though without rigorous evaluation, the quality of these generated images can still be compared. Obviously the three models can recover some visual features from encoded latent variables, however the simple autoencoder lack the ability of recovering details of an image, it tend to generate more blurry images or images contain noise; VAE and VAE with triplet loss performs better then autoencoder, though there are also very blurry images made by VAE and VAE with triplet loss. Moreover adding triplet loss to VAE seems not hurt its encoding-decoding ability much.



(a) (b) (c) (d)

Figure 5.3: Randomly selected test samples showing the encoding and decoding ability of the three models. (a) the original images; (b) reconstructed images from autoencoder; (c) reconstructed images from VAE; (d) reconstructed images from VAE with triplet loss.

Now we turn to the learned embedding space from these three models. Figure 5.4 shows that the latent codes of training data and test data. The latent codes from both datasets are projected onto 2D plane by using t-SNE method (van der Maaten & Hinton, 2008) running about 1000 iterations. The dot plots of 5.4(a), 5.4(b), 5.4(c), and 5.4(d) show that it is difficult to distinguish the categorical information about different places in the latent space. In 5.4(e), 5.4(f), the latent codes of observations form 7 clusters representing different places. Images of different places in the latent space are thus clearly distinguished and those from the same place are much closer. It is easy to reach a conclusion that without a clustering mechanism, autoencoder and VAE can hardly learn a latent space that preserves the categorial place information, while adding a triplet loss forces the VAE model to encode the categorial place information in the latent codes.

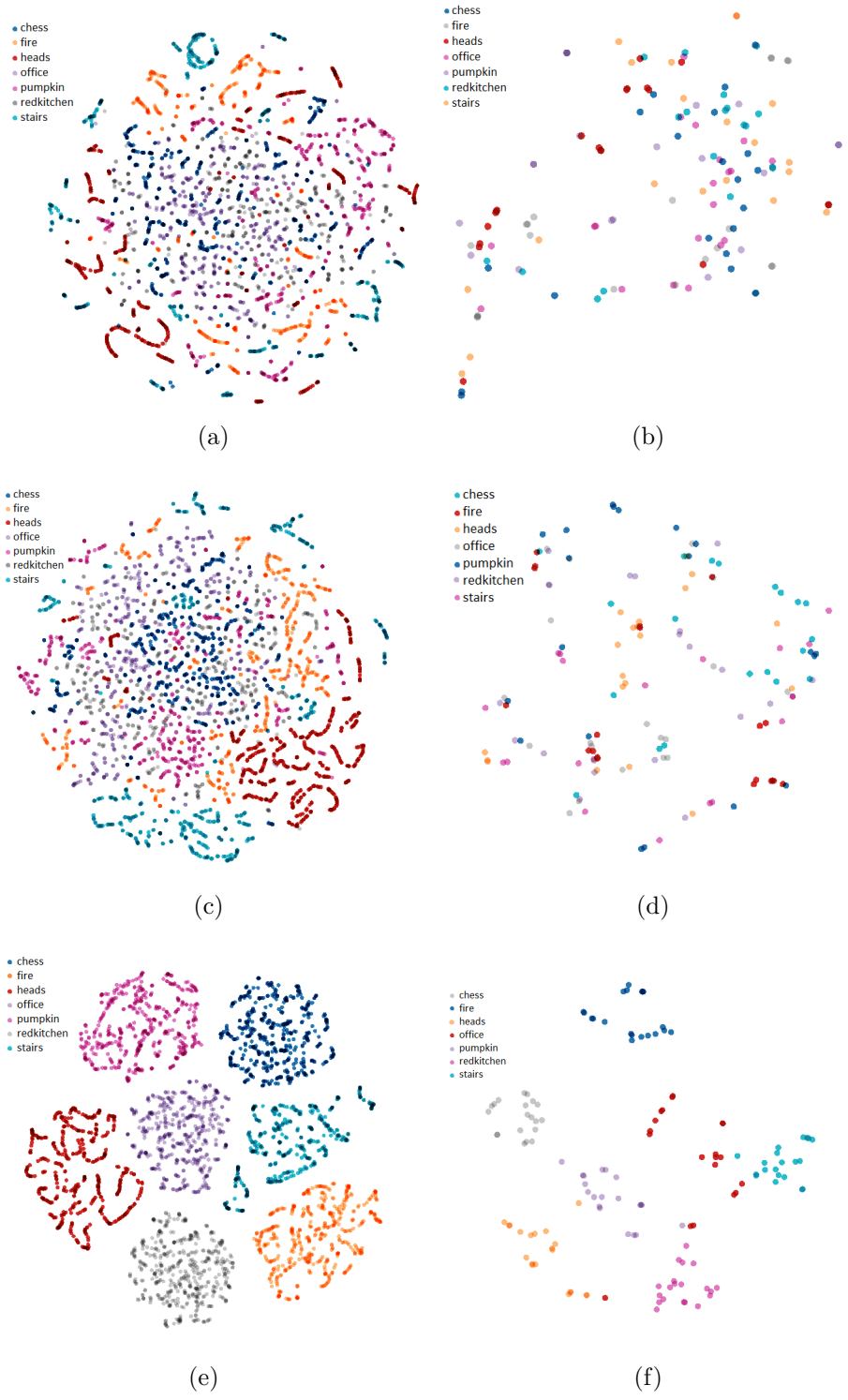


Figure 5.4: Encoded latent variables of training and test images. (a) latent codes of training image from autoencoder; (b) latent codes of test image from autoencoder; (c) latent codes of training image from VAE; (d) latent codes of test image from VAE; (e) latent codes of training image from VAE with triplet loss; and, (f) latent codes of test image from VAE with triplet loss.

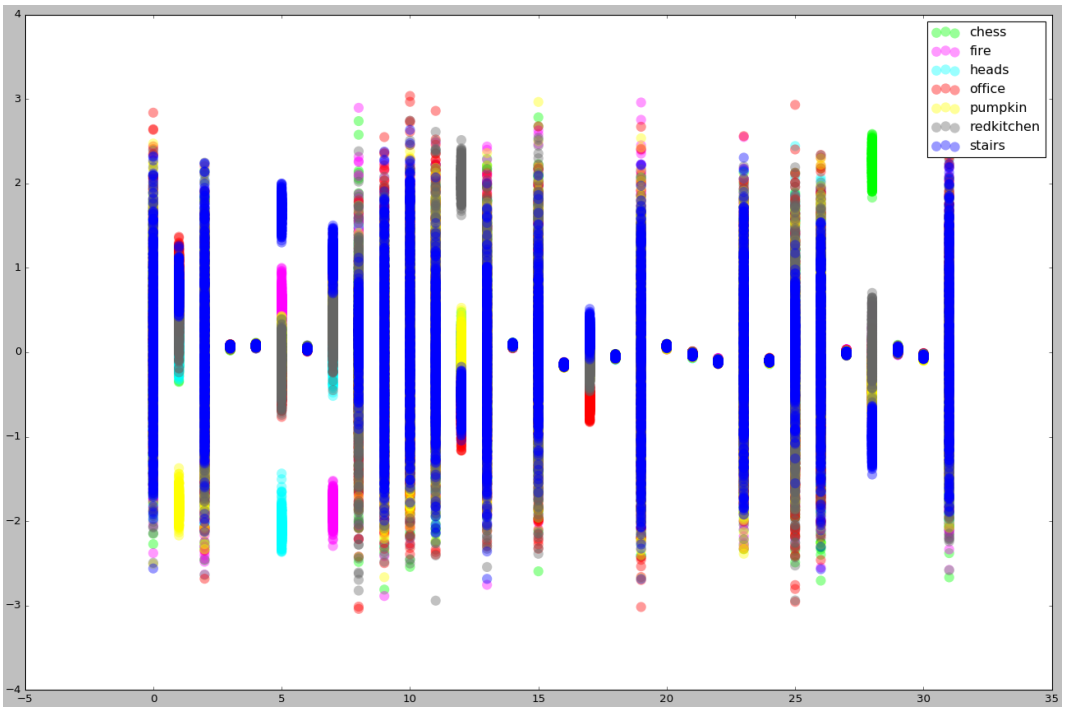
Figure 5.5 shows a scatter plot summarizing the distribution of the latent code values across dimensions. For each dimension, we plot the values from different latent codes of this dimension in different colors. In some dimensions, the values are scattered in a large range and show no relation with semantic information (i.e., place irrelevant); in some other dimensions, the place related visual properties are clearly separated (marked in black box); in the rest dimensions the values are restrained in a much smaller region near zero, which suggests little information is preserved.

5.1.5 Discussions

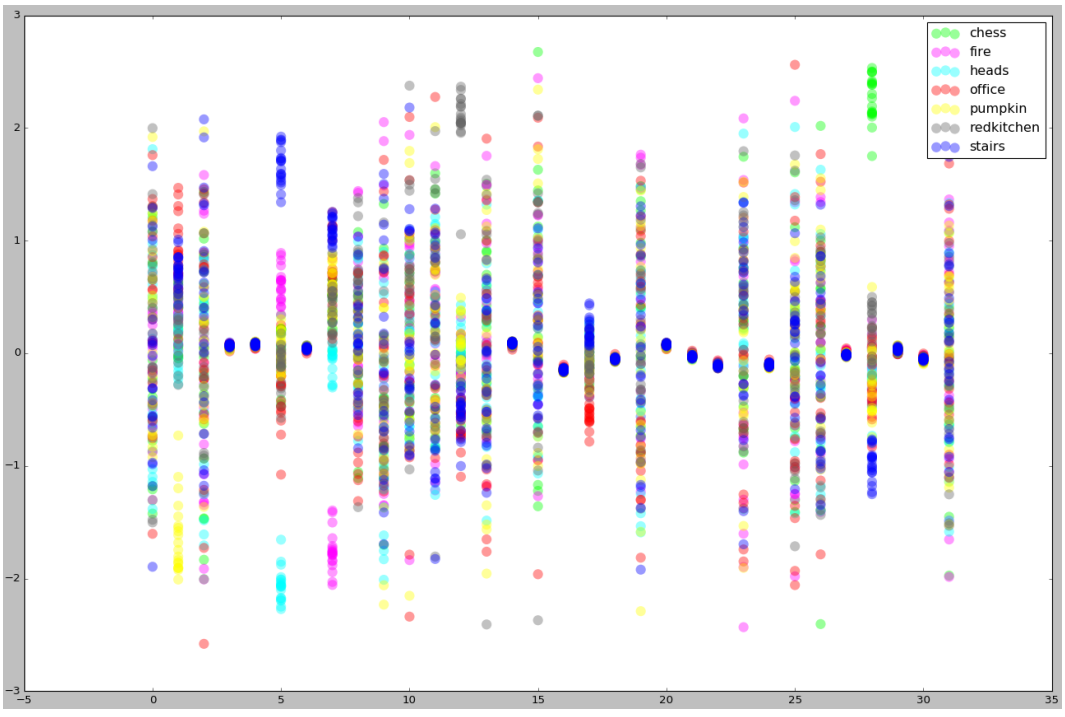
Considering the composing parameters λ_{KL} and λ_{triplet} for the KL-divergence loss and the triplet loss are small values, their contribution to the loss is much smaller than the reconstruction error. By adjusting the composing parameters, we can make a balance between knowing better the data distribution or encoding more place information in the latent codes. Through carefully selecting the composing parameters we may find two suitable parameters for the two terms in the loss function, such that the three terms in the loss can be optimized well. Similarly, other parameters can affect the final result, including the number of dimensions of the latent variable, different network structures. In this experiment, we do not search such parameters, and instead, we concern more about whether the semantic difference between places is reflected in the latent space through the VAE plus triplet loss method.

From Figure 5.3, we can see that the three models have the encoding-decoding ability, which means the learned latent spaces maintain the visual information of a place from the training images. Moreover, in generating novel images, the autoencoder is not as powerful as VAE and Triplet-VAE and adding the triplet term in VAE loss do not hurt VAE's encoding and decoding power. After adding a triplet loss term, the model still maintains the generative power to sample new images compare to the original VAE. Figure **??** shows some examples of newly sampled images that exist in neither training set nor test set. In Figure 5.4, it is illustrated clearly that without the supervision signal the autoencoder model and the original VAE cannot catch the place related information. To further investigate how the latent space learned from VAE + triplet loss preserves the place related information, we plot the distribution, extreme, mean, and medium values in Figure 5.5. It is interesting that most dimensions collapse into a tiny region near zero, while in several dimensions, their value scatters at a large range, and there is also a dimension shows a strong correlation to the place information.

The cause of this phenomenon that we observed from Figure 5.5 can be intuitively explained by examining the two KL-divergence related terms. On the one hand, the KL-divergence between recognized distribution $p(\boldsymbol{z})$ and the prior $q_\theta(\boldsymbol{z}|\boldsymbol{x})$ pushes the recognized distribution of the latent codes to a standard normal distribution, which makes most of the dimension to collapse into a tiny region near zero. On the other hand, the KL-



(a)



(b)

Figure 5.5: The scatter plot of latent code distributions across dimensions. (a) latent codes of train images, (b) latent codes of test images.

divergence can be very small if two normal distributions have similar mean and similar deviation, even the deviation is large enough, which means two individual samples drawn independently from the two normal distribution can be very different. In this sense, the dimensions in the latent space that contain place specified information are clustered into small regions, and the clusters of different places are distinguished; the information that is hardly compressed may contain uncontrollable conditions that vary among different images but not related to places.

Through this experiment, we see that the Triplet-VAE model learn a place representation that contains the visual information of places, and preserves place categories in the latent codes. Adding the triplet loss does not hurt the generative power of original VAE. It is still able to generate novel images of a different place. By altering the dimensions that preserve the categorical information, we can generate novel image observations in terms of different places. In this experiment, we do not pursue optimal hyper-parameters that can lead to a better performance (e.g., generating clearer images). This experiment proofs that when we have prior knowledge about a different place, we can adopt the triplet lose as a supervised learning way to retain categorical information in the learned place representations. In the next experiment, we will show how to use geographic distance as a weakly supervision signal when there is no prior knowledge about places.

5.2 Learning Gaussian Mixture Places Representations (GMPR)

In the previous experiment, we show the Triplet-VAE learns place representations within a single normal distribution. The results show the learned latent codes are clustered according to which place they are from. This leads to our second place representation model, the GMPR, where each place is represented as a component Gaussian distribution. In this section, we illustrate the learning process of this place model under two conditions. The pre-defined place information is known for each observation image in the first condition while in the second no pre-defined place information is available.

5.2.1 Experiment Settings

In this experiment, the network structure for Center-Triplet-VAE is similar to the one represented in the first experiment as shown in Figure 5.6. The network structure has the same encoder network as in the first experiment, which consists of 4 Conv layers and a fully connected layer (see Figure 5.1). The decoder network consists of 8 ResBlocks followed by 3 upsampling pixel shuffler layer. In the first part of the experiment, we evaluate the discriminative power of the learned representation and compare the result with other methods in terms of classification. In the classification task, the model determines from which place a query image comes. We use the classification accuracy as the performance indicator. Classification accuracy can be simply computed as Equation 5.3. We compare the performance

of Center-Triplet-VAE, Center-Triplet-VAE+softmax, CNN+softmax, center-triplet loss, center-triplet loss + softmax. The Center-Triplet-VAE+softmax adds an additional loss term to reinforce the classification power of the model. CNN+softmax uses a softmax layer after the convolutional layers to output the classification result over all classes. Center-triplet loss uses purely Euclidean distances between a class center and a latent code, while center-triplet loss + softmax adds the softmax loss to reinforce the classification power of center-triplet loss. To set up the baseline, we adopt the same structure of the encoder network as the feature extractor for all these methods. Batch hard strategy is used in all triplet relevant methods to form valid triplet groups. The training environment remains unchanged as the first experiment.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{5.3}$$

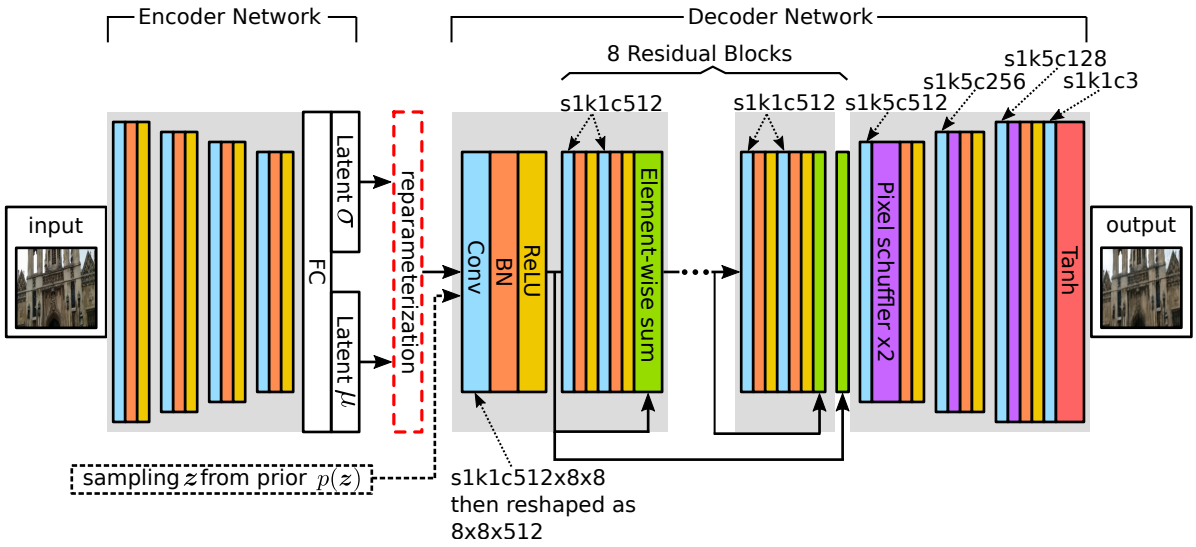


Figure 5.6: The network architecture used for Center-Triplet-VAE model. The encoder network is similar to the one used in the first experiment. The decoder network is altered by using residual blocks and pixel shuffler layer.

5.2.2 Data

In the first part of the experiment, we still use the 7 scenes data the same as in the first experiment (see Figure 5.2 for an overview of this dataset). We randomly select 1,000 images from each scene to form a training dataset. For evaluation, we randomly select another 500 images from each scene to form a testing dataset.

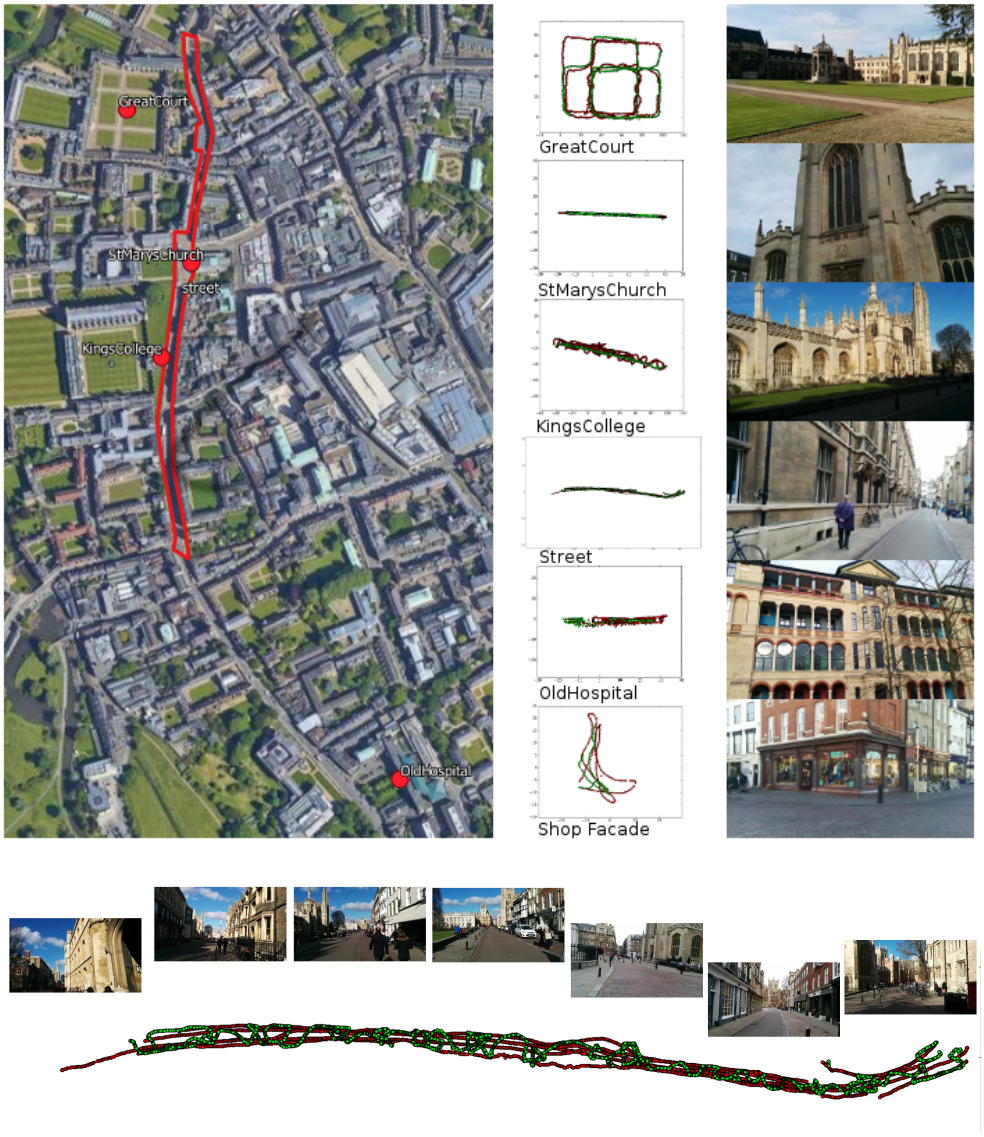


Figure 5.7: Overview of Cambridge Landmarks Dataset and the "Street" subset. The red dots stand for the camera positions of training images, while green dots are for the test images. We plot all dots in their local reference frame. (a) The whole dataset; (b) The "street" subset.

In the second part of the experiment, Cambridge Landmarks Dataset is used. This data set is first proposed to evaluate neural network based methods for urban camera relocalization (Kendall et al., 2016). The data set contains more than 12,000 images labelled with their 6DoF camera pose captured from 6 scenes around Cambridge University. Different from the Microsoft 7 scenes dataset, it covers a large outdoor environment with complex visual features and noise, such as vehicles and pedestrians. The dataset consists of 6 outdoor scenes. The "street" subset contains images along St. Johns Street - Trinity Street - King's Parade - Trumpington Street while the images of the other scenes record one salient building and its surrounding environment including the Great Court of Gonville & Caius College, University of Cambridge, St Mary's Church, King's College, Old Hospital, and a shop facade at street corner. In Figure 5.7(a) we give an overview of this data set, where the left part shows the geographic region, the middle part is a list of data samples, and the right side shows an example image for each scene. The dataset is originally partitioned into a training set and a testing set as well. The training samples are marked as red dots, and the test samples are green dots (see the upper middle part of Figure 5.7(a)). The numbers of images are seriously imbalance across the scenes. The smallest dataset contains only about 400 images in training set and 300 images in the test set, while the "street" contains the largest number of images, which is about 3,000 for training and another 3,000 for testing.

5.2.3 Results of Learning GMPR with Pre-Defined Places

To evaluate the discriminative power of the learned latent representation by the proposed Center-Triplet-VAE, we conduct extensive experiments on various methods, including CNN+softmax, center-triplet loss, center-triplet loss + softmax and center-triplet-VAE on the 7 scenes dataset. In this experiment, we set all hyperparameter lambda to 0.02.

We can see from Tab. 5.3, CNN+softmax performs the best on the 7 scenes dataset, while center-triplet-vae and center-triplet follow tightly behind. The center-triplet loss + softmax performs not so good respectively. Given the condition that we do not particularly tune the hyperparameters to squeeze the performance of all these model, we can hardly say which model is better. However, all the models except for the center-triplet loss+softmax model show their power and robustness (not sensitive to the value of hyperparameters) in this classification task.

Table 5.3: The performances (%) of different methods on 7 Scenes dataset.

methods	accuracy
CNN+softmax	96.17
center-triplet loss	94.086
center-triplet loss + softmax	82.8
center-triplet-vae	95.03

To visually explore the property of the learned latent codes, we adopt PCA (principal component analysis) to visualize the latent codes on the 7 scenes dataset. As is shown in Figure 5.8, some interesting properties can be observed, (1) the learned latent codes of the same place form clusters by all these methods; comparing with softmax loss, the triplet-based methods produce distinguishable clusters of learned latent codes from the same place; (2) the proposed Center-Triplet-VAE performs better than "center-triplet loss" and "center-triplet loss + loss" on achieving small intra-class variance and large inter-class variance; (3) the latent codes that are produced by Center-Triplet-VAE locate around its center, while the latent codes that are produced by center-triplet loss show an "angle-edge phenomenon". The "angle-edge phenomenon" is caused by the inner property of triplet loss function. The function pushes the latent codes of different places to opposite directions, but as long as the distance exceeds the margin value, the loss function provides no information to the optimization process. In Marc-Olivier Arsenault's blog, he analyses this phenomenon and proposes an improved version of triplet loss[1]. Figure 5.8 show the latent codes of test data.

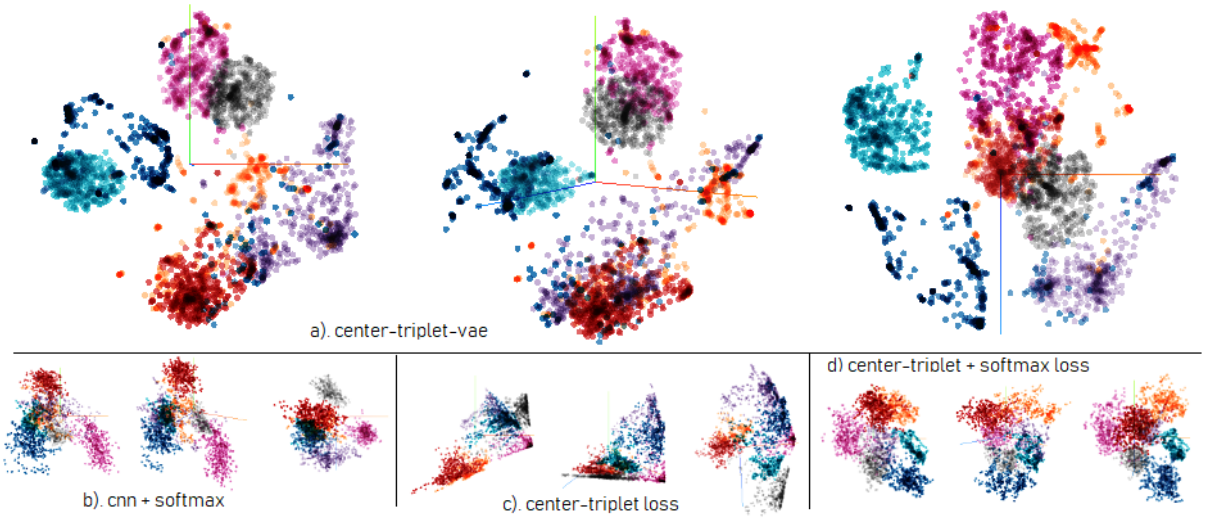


Figure 5.8: The learned latent codes of test data in 7 Scenes dataset. Each visualization consists of front, top and side view from left to right.

In Figure 5.9, we give some example images that are generated directly from the latent codes. The latent codes are sampled from each place representations, i.e., Gaussian components in the same way of reparameterization. Some very nice images are reconstructed from these randomly generated latent codes, while there are also very blurry images. Though some of these images are very blurry, it is still possible to tell some visual features that are relevant to the place from these images, e.g., texture and color.

[1]Lossless Triplet loss. https://towardsdatascience.com/lossless-triplet-loss-7e932f990b24 access date:22.11.2018

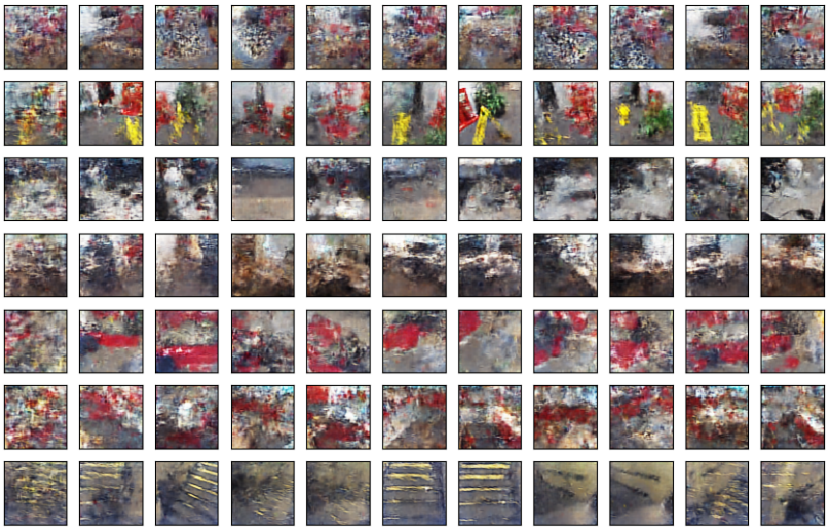


Figure 5.9: Examples of images sampled from randomly generated latent codes. The first column shows the images generated from the latent codes of mean value from each Gaussian component.

The origin-reconstruction figure (Figure 5.10) are very interesting. The decoder successfully recovers some images and left some with small faults. Some of the recovered heads images are mixed in some red and yellow textures, which may come from the fire or kitchen data. Overall speaking, the quality of generated images is not as good as the Triplet-VAE. However, considering we double the training data amount in this experiment, it could be much harder for the encoder to learn the data distribution. Therefore, we assume that a more powerful encoder can improve the generated image quality.

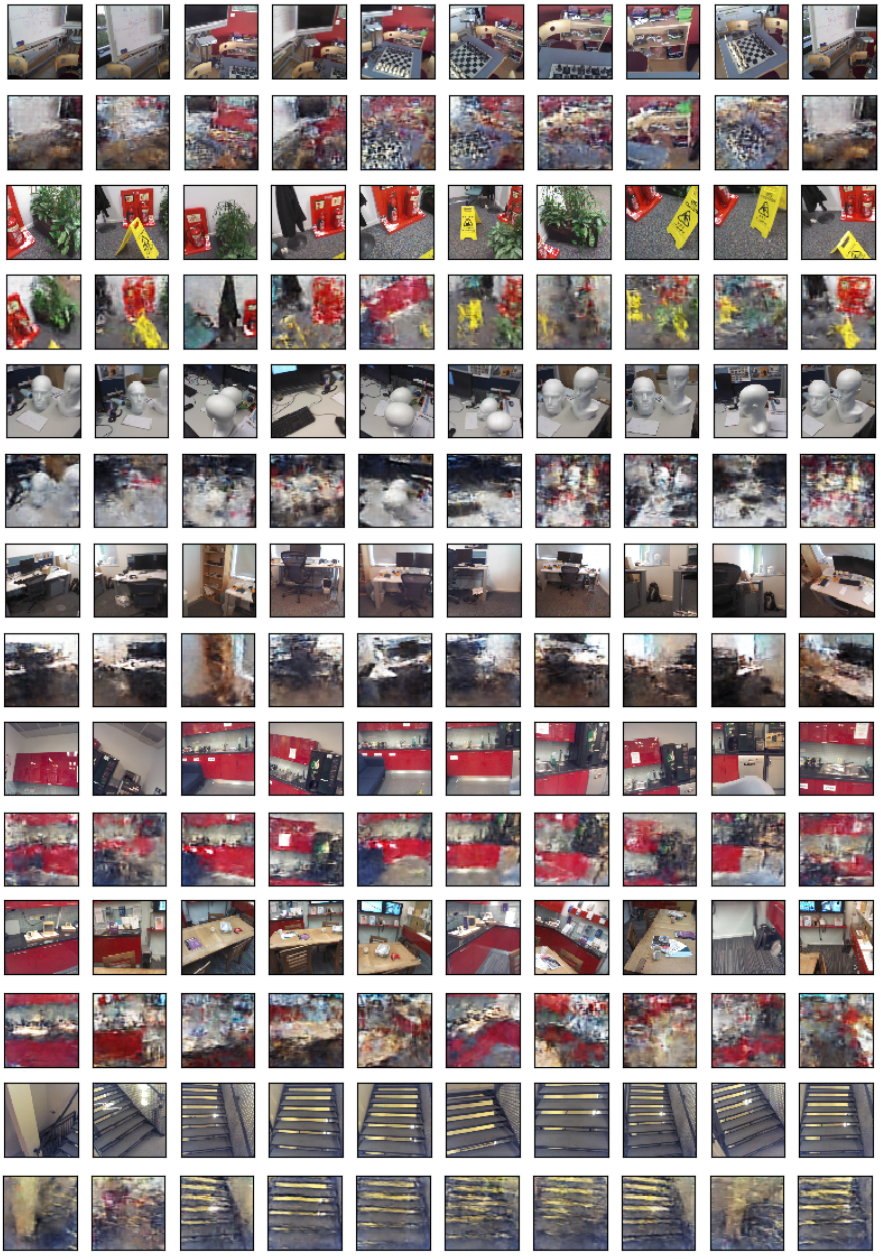


Figure 5.10: Examples of decoding results. Every two rows represent images of the same scene. The odd rows show the original images, and the even rows show the reconstructed images.

5.2.4 Results of Learning GMPR without Pre-Defined Places

In this experiment, weakly supervised Center-Triplet-VAE is used to learn a GMPR without pre-defined place information. The available information includes the image observations themselves and their corresponding geo-locations (from their camera poses). The geo-locations act as weakly supervision signal in training phase. However, we still need to specify the number of places as a prior knowledge to set the number of Gaussian components in GMPR. The supervision signal is generated according to the geo-location of images. For each training sample, the anchor, we gather a set of potential positive samples whose distances to the anchor are within a certain value (referred as PosDist). Meanwhile a set of definite negative samples are collected with all their distances to the anchor is greater than a certain value (NegDist). It is worth noticing that the PosDist and NegDist are largely restrained by the number of assumed number of places. Too large PosDist can cause too many samples from different places in the possible potential positive group and lead to computational inefficiency. Similarly, too small NegDist can mix in some positive samples in the definite negative group. Too small PosDist or NegDist can reduce the choice when forming the triplet groups. To decide PosDist and NegDist we assume places centers and their distances as reference. Assume the n number of place evenly divide the space into n coverages, each with a center that every location in the place has the smallest distance to the center. Given these centers, we can calculate the smallest distance between these centers, and we choose 1/3 of this distance as PosDist and itself as the NegDist.

The following figures show the learned place representation on the "street" in Cambridge Landmark dataset. We assume images of this dataset come from 6 places and estimate suitable PosDist and NegDist to form the triplet group. Figure 5.11 show the classification result of the images with 6 learned places, points in the same color indicate that they belong to the same place. It is worth to point out that we actually get each image observation 6 values of the possibility to assign it to the 6 places, but when we decide from which place it comes from, we simply choose the one with the largest possibility.

The learned latent codes is visualized in Figure 5.12 with PCA dimension reduction. It shows the image observations from places are distinguished clearly in the latent space, though some of them are geographically close to each other.

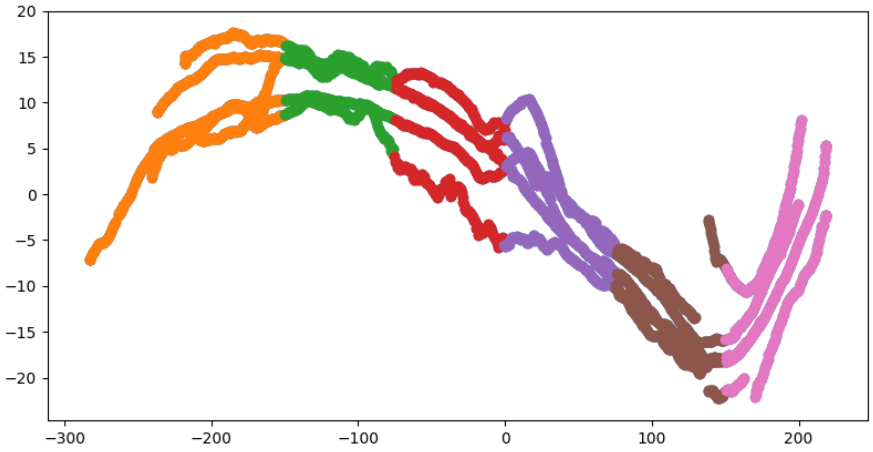


Figure 5.11: The place classification result of the "street" images. Each image is assigned with a color indicating its place class.

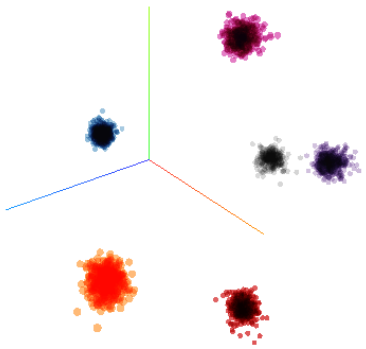


Figure 5.12: The learned latent codes of 6 places on "street" dataset visualized by PCA dimension reduction method.

5.2.5 Discussions

In this experiment, we show how to learn a GMPR under two conditions. When pre-defined place information is given, we can easily build the correlation between a single image observation and its corresponding place. A binary mixture coefficient can be introduced according to this correlation. The learning problem can be tackled as simple as we separately estimate the parameters (mean and variance) for each place. In this case, the GMPR is not actually a mixture of distributions. The proposed Center-Triplet-VAE can easily learn such a model and enable the discriminative power to distinguish the observations from different places. The discriminative power is important to extend the applications of the learned latent codes, such as place recognition.

When pre-defined place information is not given, no explicit correlation between place and their observations can be found. A continuous variable is used as a mixture coefficient to indicate the possibility of assign an observation image to a specific place. Estimating the mixture coefficient, the GMPR parameters, and the parameters of the encoder network are particularly difficult to facilitate learning and enhance the geographical relationships between image observations, we turn to the geo-locations as a weakly supervision signal. These experiments show that the learned place representations, on the one hand, preserve the visual information in the latent codes, and on the other hand show strongly geographical correlations between observations.

The experiment results also show that the latent space may be very sparse, as it is hard to reconstruct realistic images from some of the randomly generated latent code. The reason could be two folds. One is that the encoder is not powerful enough to approaching the true data distribution. The other is the training data samples are sparse as well. If we deal with the place representation learning problem at a much larger scale, on the one hand, we may need more powerful networks (deeper and more advanced functional structure), on the other hand, we can employ some data enhancement techniques to expand the amount of training data.

5.3 Learning Places Representations with Camera Poses

5.3.1 Experiment Settings

In this experiment, the network structure for conditional VAE in shown in Figure 5.13. We use a similar encoder network as in the previous two experiments. The only change is to add an auxiliary Conv block to combine the image input and the camera pose input. The Conv block first transforms the camera pose with a convolutional layer (kernel size =1, strides = 1) followed by a batch normalization layer and a ReLU activation layer. Here the Conv block acts as a fully connected layer. Then the transformed camera pose is reshaped into the same size as the feature map from the first Conv layer that takes images as input. The output of the encoder network consists of three parts, the mean and variance of the uncontrollable latent variable and the estimated parameters of the latent place representation. We use the mean value of the estimated normal distribution to approximate the parameters of the latent place representation (i.e., the mean and variance of a Gaussian distribution). In this experiment, we set the variance of the latent place representation as a fixed value. An empirical experiment is given in the next part showing the impacts of different values. The decoder network is the same as shown in Figure 5.13. The input of the decoder network is the concatenation of the latent variable of the uncontrollable condition, the latent place representation, and the camera pose. The red dashed lines show the reparameterization trick to sample the latent variable $\boldsymbol{z}$ of the uncontrollable conditions from the output $\boldsymbol{\mu_z}$ and $\boldsymbol{\sigma_z}$, and the estimation of the mean $\boldsymbol{\mu_r}$ for the latent place representation. In the inference stage, we sample from the standard normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, i.e., the prior distribution of $p(\boldsymbol{r})$, for the uncontrollable latent variable, and use the estimated $\boldsymbol{\mu_r}$ together with a specific camera pose to generate a desired image observation.

The rotation parameter of the camera pose is represented as a quaternion of 4 parameters. The 4 rotation parameters together with 3 translation parameters are then fed into the network. The hyper-parameters and training details are listed in Table 5.4. The training parameters are similar to the previous two experiments. As the KL term that measures the difference between the estimated distribution and the prior of the latent place representation has been added in the loss function, add a new coefficient, lambda ($\mathrm{KL}_{\boldsymbol{r}}$) for this term. Another new hyper-parameter is the variance of the latent place representation $\mu_{\boldsymbol{r}}$, which is fixed as 0.01. The training environment is the same as Table 5.2 shows.

Table 5.4: Training Parameters

lambda(KL_z)	lambda(KL_r)	learning rate	batch size	epochs
0.002	0.002	0.001	128	6000
weight init	μ_z dim	μ_r dim	σ_r	-
N(0, 0.02)	8	16	0.01	-

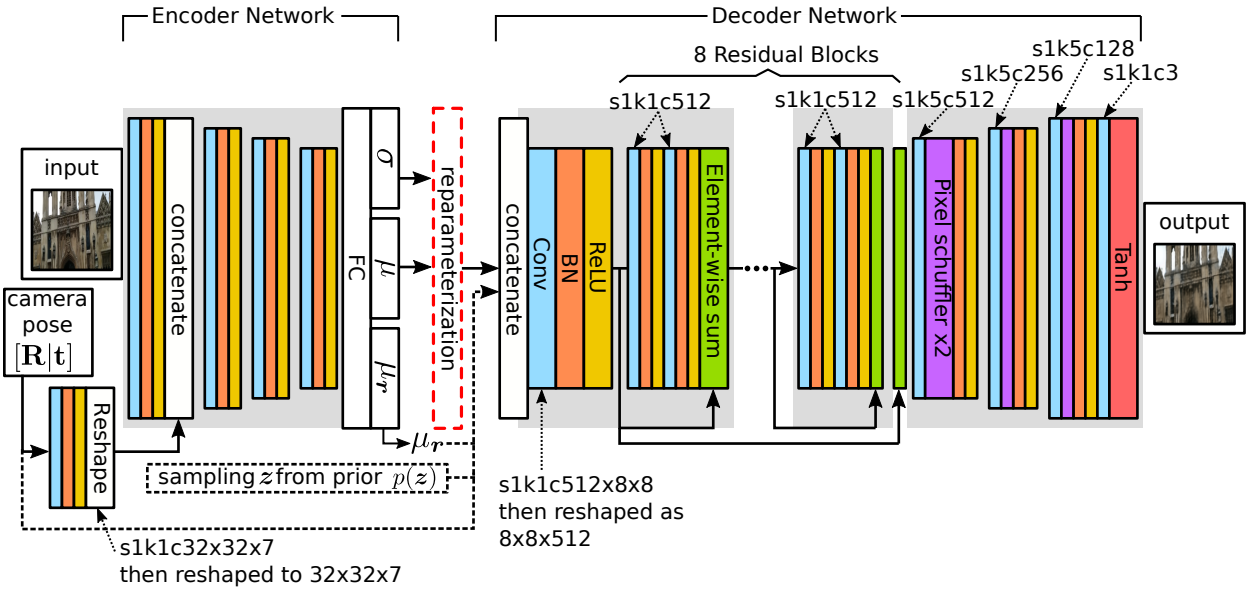


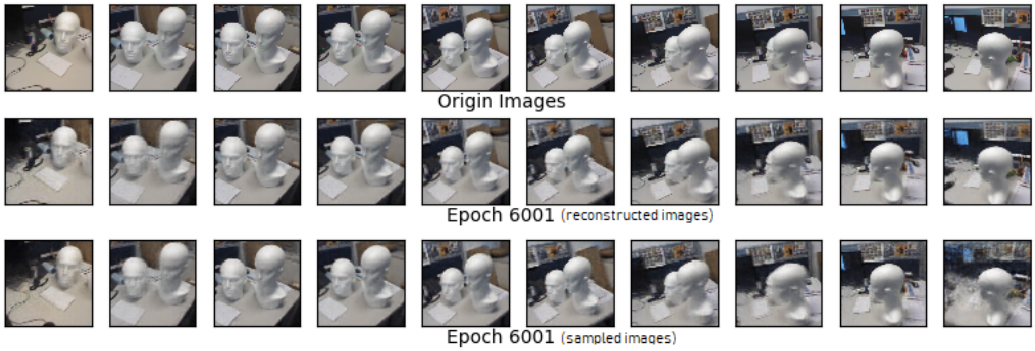
Figure 5.13: The network architecture with extra camera pose component and a latent variable for place representation.

5.3.2 Data

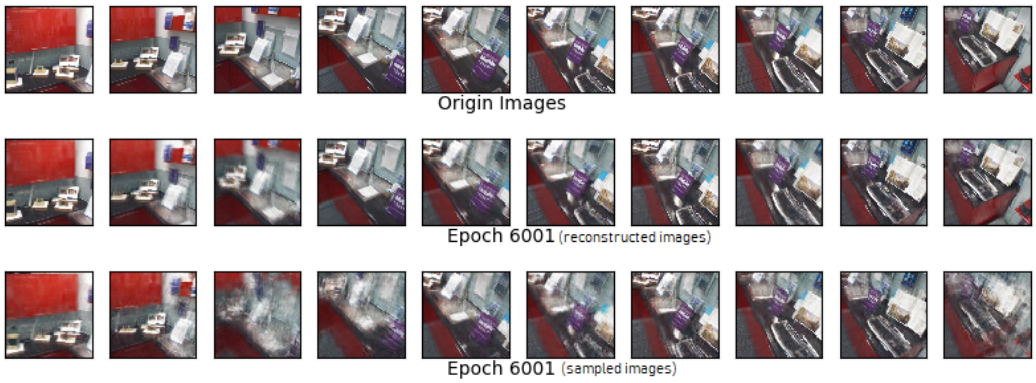
We select 5 image sequences from both the 7 scenes dataset and the Cambridge Landmarks dataset to test our implementation. We select 3 sequences from "heads", "red-kitchen", and "office" in the 7 scenes dataset. Another 2 sequences are from "the great court" and "St.Mary's church" in the Cambridge Landmarks dataset. Each image sequence consists of 200 images, and about 50 images are selected evenly from the sequence for test purpose. The rest images are used for training

5.3.3 Results

In the following figures (Figure 5.14), we show the result of each image sequences. In each figure, 10 of the original test images are given in the first row, followed by a row of reconstructed images and a row of sampled images. The reconstructed images are very similar to the original test images. The sampled images are generated with the latent place representation, randomly sampled latent uncontrollable conditions, and the same camera pose of the original test images. Except for some details, the sampled images contain the major visual information of a place. However, some of the generated images show a slight change in the viewpoint. Figure 5.15 show the learned place representation from the five image sequences. The 5 place representations show their unique patterns across dimensions.


Origin Images
Epoch 6001 (reconstructed images)
Epoch 6001 (sampled images)

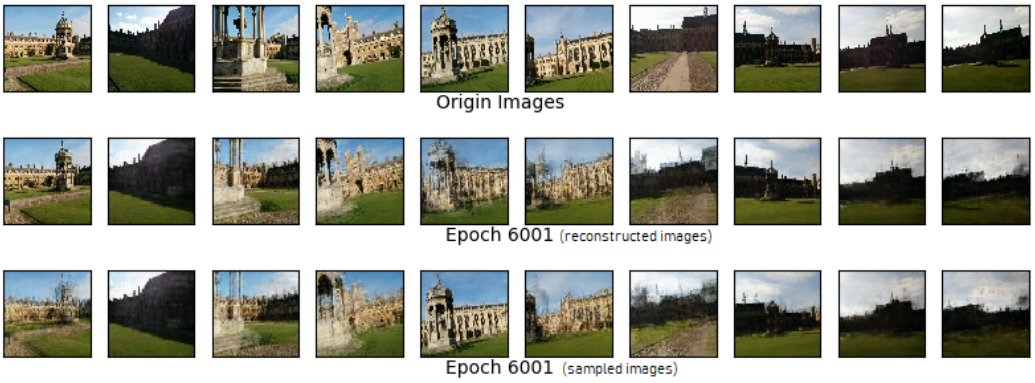
(a)


Origin Images
Epoch 6001 (reconstructed images)
Epoch 6001 (sampled images)

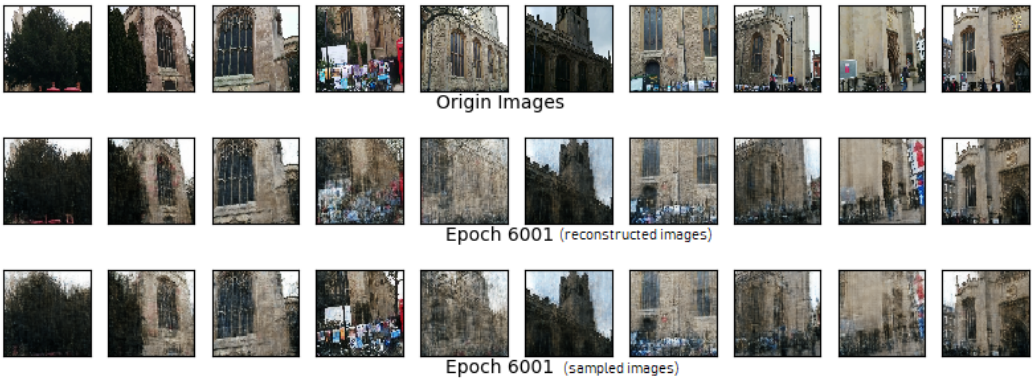
(b)


Origin Images
Epoch 6001 (reconstructed images)
Epoch 6001 (sampled images)

(c)



(d)



(e)

Figure 5.14: Original images, reconstructed images, and the sampled images for the selected 5 image sequences. (a) "heads", (b) "redkitchen", (c) "office", (d) "the Great Court", (e) "St Mary's church".

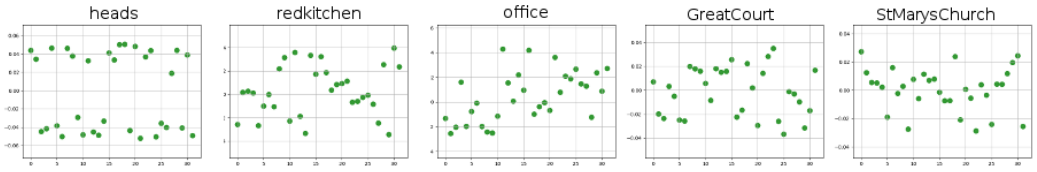


Figure 5.15: The learned latent place representations that are expended along the latent dimensions.

5.3.4 Discussions

In this experiment, we fixed the variance in the prior of the latent place representation, and thus introduce one more hyperparameter. By altering this value, we can control the range of possible values in the latent place representation. It is worth to point out that theoretically when the variance is set to 0, the normal distribution degenerates to a deterministic value. Usually, we expect the variance to be small, which intuitively means that there is less uncertainty in the place representation. However, how different variance value influences the performance of the model is unclear. We run a small scale empirical experiment to address this issue. We set 5 different variance values (0.01, 0.1, 0.5, 1, 5) and test these values on the "heads" sequence. The results (Figure 5.16) show little difference in generating novel images among these values. However, using a larger variance leads to faster converge than using a smaller variance, while a too large variance also needs more time to converge. The smaller variance value usually causes large KL divergence term and squeezes the estimated mean together, which may lead to difficulties in optimization, while a too large variance may weaken the representation ability of the latent variable. Since our data is relatively small, the network may be too powerful to overcome these issues. There is another possible implementation where the variance of the latent place representation is added as a learnable parameter in the same way of the mean. Though a place has its geographical boundary such that the range of possible camera pose values is limited, our image observations are still very sparse comparing to the limited range. The sparse observations only provide incomplete visual information about a place from specific viewpoints. The sampling images of the results show that the proposed method does learn visual properties from image observations. However, these camera poses are close to the camera poses of the training images, which leaves the question "to what extent the learned visual properties can be used to infer the appearance of the unobserved part of a place?" open. Another issue may rise in terms of the "resolution" of the camera pose in a learned model as we notice that there are some sampled images showing contents under slightly rotated and translated camera poses compare to the original test images.

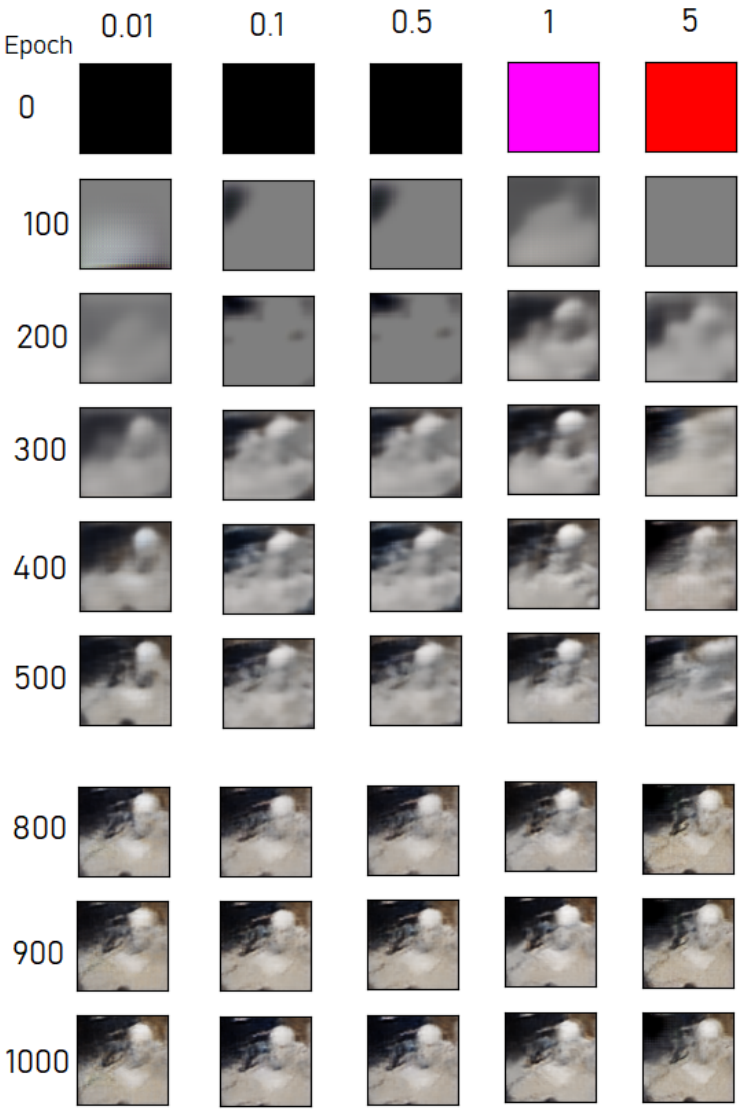


Figure 5.16: An example of a sampled image in the first 1000 steps (100 steps interval) with 5 different variance values, 0.01, 0.1, 0.5, 1, and 5.

Chapter 6

Conclusion

6.1 Summary

In this dissertation, we addressed the challenge of understanding a place by detecting visual and spatial(semantic) properties from voluntarily collected images. Starting from a framework of place learning from a collective image dataset, we proposed two representations of place, which are based on the latent variable model to encode visual property into place representations. By introducing a generative model, specifically variational autoencoder (VAE), the latent place representations can be jointly learned with the distribution of image observations. In the first place representation, spatial information of the image observations is implicitly injected in place representations during learning. The spatial property of a place is not explicitly disentangled from the visual property of a place as well. The second place representation is disentangled with visual property from camera pose. Based on the two place representations, we performed three experiments to show possible ways to encode visual and spatial information in latent place representations. We model the first place representation as a probability distribution. In the first version, we assume a single normal distribution can represent all the observations in the latent space. Correspondingly in the first experiment, we adopt triplet loss and incorporate categorical(semantic) information to learn the latent place representation. The results show that the latent codes retain the visual information in the images; some of the dimensions in the latent representation show a strong correlation with their place labels. The results of the first experiment lead us to the second version, a GMM-based place representation (we refer it as GMPR), where each place is represented as a normal distribution component and the latent space in a mixture of these distributions. In the second experiment, we learn such a GMPR under two situations, 1). the places are pre-defined and 2). no explicit place information is available. We use Center-Triplet-VAE to learn GMPR. Especially, we adopt geo-locations as a weakly supervision signal in the second situation to facilitate place learning. The learned latent representation connects the spatial and visual property of a place with a certain probabilistic distribution. The latent codes and place representations are feasible for discrete place recognition. To disentangle the visual and spatial property,

we introduced holistic place representation and a conditional VAE model. In the third experiment, we use the conditional VAE to learn the holistic latent place representation without the impact of camera pose and other conditions. Then the learned place representation together with camera pose and other conditions can be used to generate novel images with designated camera poses.

This work begins with the latent variable model and deep learning methods. We represent the visual and spatial properties of a place with latent variables. VAE, as an extensively investigated generative model in recent years, is used to learn the relationship between visual and spatial properties from voluntarily collected images. Comparative methods are used to ensure that the image observations of different places are distinguishable in the latent space. We summarize our main contributions as the following,

1. We have extended the understanding of place in GIScience from a data-driven perspective. We have proposed probabilistic latent place representations, which links the visual property of a place and visual observations. The place representations can be enriched by involving semantic and spatial information in the learning process so that the representations of multiple places contain an implicit map of the environment. The visual place representation is potentially helpful in querying the place related information given images as the query condition.

2. We have proposed a specific framework to approach a collective place definition and introduced deep learning methods, comparative learning, and VAE, to implement this framework. The proposed methods include adding triplet loss and weakly supervised triplet loss as a regularization term to the VAE loss function and using a conditional version of VAE to involve camera pose in the learning process directly. By adding triplet loss, semantic information of distinct places is preserved in the latent representation. With weakly supervised triplet loss, geographically close images are maintained close in the latent space. By involving camera pose in the learning process, we show the possibility to synthesize a novel image about a place given a new camera pose.

3. We have shown the possibility of connecting human perception of spatial knowledge and the machine perception of spatial knowledge with the proposed place definition and corresponding computational models. On the one hand, the place representation increases the accessibility of spatial information for human users; on the other hand, the proposed place representation provides a way of interpreting how machine perceives platial information.

6.2 Outlook

Understanding the visual and spatial properties is the first step to approach the concept of a place. There is far richer information other than visual and spatial properties attached

to a place. Place as a spatial container also bears various spatial-temporal phenomenon, such as human emotional responses and social activities. In the very beginning of this dissertation, we mentioned some researches about combining the visual properties of a place to spatial-temporal phenomena. In these researches, supervised learning methods are used to learn task dependent visual properties. This discriminative learning schema can hardly capture structural information between the visual properties and the target phenomena of a place. By introducing generative methods (e.g., GANs and VAEs), it is possible to learn a latent representation of the visual property while encodes the target phenomenon in the latent representation at the same time. Moreover, there are more and more data sources available in recent years, especially about the urban area in the context of smart cities. These data source also provide valuable information about a place. The feature directions on this topic can be drawn as the following,

- The development of theories and tools to encode a specific kind of spatial-temporal phenomenon into a latent representation as well as the methods to combine the visual properties and the spatial-temporal phenomenon.
- The application domain of the learned latent representation. It is about the scope and the usage of the latent representation to support further deep learning tasks dedicated to more complex spatial intelligent problems.
- The development of theories and tools to take the advantages of rapidly growing data sources. It is about how to extend the existing generative models to learning place representations from multiple data sources.

With regard to the challenge of combining the visual and spatial properties about a place itself, we may ask how to distinguish different places from the latent representation and what can be the application scenarios of the latent place representation. Solving the first question helps to answer the second question. Take the autonomous driving as an example, the image observations are consecutively captured, and these images may contain visual properties of a different place. Extract and distinguish the information from mixed image observations can improve the understanding of place, and enable us to model broader geographic space. The model of multiple places will help us to develop interpretable spatial cognition models for machines, which can be used to support complex spatial intelligent behaviors.

Bibliography

Adibi, S. (Ed.). (2015). *Mobile Health* (Vol. 5). Cham: Springer International Publishing. doi: 10.1007/978-3-319-12817-7

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., & Szeliski, R. (2009, sep). Building Rome in a day. In *2009 ieee 12th international conference on computer vision (iccv)* (pp. 72–79). IEEE. doi: 10.1109/ICCV.2009.5459148

Alazzawi, A. N., Abdelmoty, A. I., & Jones, C. B. (2012, feb). What can I do there? Towards the automatic discovery of place-related services and activities. *International Journal of Geographical Information Science*, *26*(2), 345–364. doi: 10.1080/13658816.2011.595954

Angeli, A., Filliat, D., Doncieux, S., & Meyer, J. A. (2008, oct). Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, *24*(5), 1027–1037. doi: 10.1109/TRO.2008.2004514

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In *The ieee conference on computer vision and pattern recognition (cvpr)* (pp. 5297–5307). doi: 10.1109/TPAMI.2017.2711011

Arjovsky, M., & Bottou, L. (2017, jan). Towards Principled Methods for Training Generative Adversarial Networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M., Chintala, S., & Bottou, L. (2017, jan). Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

Armagan, A., Hirzer, M., Roth, P. M., & Lepetit, V. (2017, jul). Learning to Align Semantic Segmentation and 2.5D Maps for Geolocalization. In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 4590–4597). IEEE. doi: 10.1109/CVPR.2017.488

Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., & Pollefeys, M. (2010). Handling urban location recognition as a 2d homothetic problem. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision – eccv 2010* (pp. 266–279). Berlin, Heidelberg: Springer Berlin Heidelberg.

Baatz, G., Saurer, O., Köser, K., & Pollefeys, M. (2012). Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In A. Fitzgibbon et al. (Eds.), *Computer vision – eccv 2012* (pp. 517–530). Berlin, Heidelberg: Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-33709-3_37

Bansal, M., & Daniilidis, K. (2014, jun). Geometric Urban Geo-localization. In *2014 ieee*

conference on computer vision and pattern recognition (cvpr) (pp. 3978–3985). IEEE. doi: 10.1109/CVPR.2014.508

Bansal, M., Sawhney, H. S., Cheng, H., & Daniilidis, K. (2011). Geo-localization of street views with aerial image databases. In *Proceedings of the 19th acm international conference on multimedia - mm '11* (p. 1125). New York, New York, USA: ACM Press. doi: 10.1145/2072298.2071954

Bao, J., Chen, D., Li, H., & Hua, G. (2017). CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training. *arXiv preprint arXiv:1703.10155*.

Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O'Keefe, J., ... Burgess, N. (2006). The Boundary Vector Cell Model of Place Cell Firing and Spatial Memory. *Reviews in the neurosciences*, *17*(1-2), 71–97. doi: 10.1515/REVNEURO.2006.17.1-2.71

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008, jun). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359. doi: 10.1016/j.cviu.2007.09.014

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. In *European conference on computer vision 2006* (Vol. 3951 LNCS, pp. 404–417). Springer, Berlin, Heidelberg. doi: 10.1007/11744023_32

Bennett, B., & Agarwal, P. (2007). Semantic Categories Underlying the Meaning of Place'. In S. Winter, M. Duckham, L. Kulik, & B. Kuipers (Eds.), *Spatial information theory. cosit 2007. lecture notes in computer science, vol 4736.* (pp. 78–95). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-74788-8_6

Boccara, C. N., Sargolini, F., Thoresen, V. H., Solstad, T., Witter, M. P., Moser, E. I., & Moser, M.-B. (2010, aug). Grid cells in pre- and parasubiculum. *Nature Neuroscience*, *13*(8), 987–994. doi: 10.1038/nn.2602

Bottou, L., Curtis, F. E., & Nocedal, J. (2016, jun). Optimization Methods for Large-Scale Machine Learning.

Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., & Rother, C. (2017, jul). DSAC Differentiable RANSAC for Camera Localization. In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 2492–2500). IEEE. doi: 10.1109/CVPR.2017.267

Brahmbhatt, S., Gu, J., Kim, K., Hays, J., & Kautz, J. (2017, dec). Geometry-Aware Learning of Maps for Camera Localization.

Brejcha, J., & Čadík, M. (2017, aug). State-of-the-art in Visual Geo-Localization. *Pattern Analysis and Applications*, *20*(3), 613–637. doi: 10.1007/s10044-017-0611-1

Bromley, J., Guyon, I., LeCun, Y., Sackinger, E., & Shah, R. (1993). Signature Verification using a Siamese Time Delay Neural Network. In *Advances in neural information processing systems* (Vol. 6, pp. 669–688). Morgan Kaufmann Publishers Inc.

Burgess, N. (2008, mar). Spatial Cognition and the Brain. *Annals of the New York Academy of Sciences*, *1124*(1), 77–97. doi: 10.1196/ANNALS.1440.002

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Reid, I., ... Leonard, J. J. (2016, dec). Past, Present, and Future of Simultaneous Localization and Mapping : Toward the Robust-Perception Age. *Transactions on Robotics*, *32*(6), 1309–1332.

doi: 10.1109/TRO.2016.2624754

Cadena, C., Gálvez-López, D., Tardós, J. D., & Neira, J. (2012, aug). Robust place recognition with stereo sequences. *IEEE Transactions on Robotics*, *28*(4), 871–885. doi: 10.1109/TRO.2012.2189497

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision eccv 2010.* (pp. 778–792). Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-15561-1_56

Castaldo, F., Zamir, A., Angst, R., Palmieri, F., & Savarese, S. (2015, dec). Semantic Cross-View Matching. In *2015 ieee international conference on computer vision workshop (iccvw)* (pp. 1044–1052). IEEE. doi: 10.1109/ICCVW.2015.137

Chen, D. M., Baatz, G., Koser, K., Tsai, S. S., Vedantham, R., Pylvanainen, T., ... Grzeszczuk, R. (2011, jun). City-scale landmark identification on mobile devices. In *2011 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 737–744). IEEE. doi: 10.1109/CVPR.2011.5995610

Chen, Z., Lam, O., Jacobson, A., & Milford, M. (2014, nov). Convolutional Neural Network-based Place Recognition. In *Proceedings of the australasian conference on robotics and automation (acra 2014).*

Clark, R., Wang, S., Markham, A., Trigoni, N., & Wen, H. (2017, jul). VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 2652–2660). IEEE. doi: 10.1109/CVPR.2017.284

Costante, G., Mancini, M., Valigi, P., & Ciarfuglia, T. A. (2016, jan). Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation. *IEEE Robotics and Automation Letters*, *1*(1), 18–25. doi: 10.1109/LRA.2015.2505717

Couclelis, H., Golledge, R. G., Gale, N., & Tobler, W. (1987, jun). Exploring the anchor-point hypothesis of spatial cognition. *Journal of Environmental Psychology*, *7*(2), 99–122. doi: 10.1016/S0272-4944(87)80020-8

Cresswell, T. (2014). *Place : a short introduction.* Wiley-Blackwell.

Cummins, M., & Newman, P. (2009). Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings robotics: Science and systems (rss).*

Cummins, M., & Newman, P. (2011, aug). Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal of Robotics Research*, *30*(9), 1100–1123. doi: 10.1177/0278364910385483

Davidson, P., & Piche, R. (2017). A Survey of Selected Indoor Positioning Methods for Smartphones. *IEEE Communications Surveys & Tutorials*, *19*(2), 1347–1370. doi: 10.1109/COMST.2016.2637663

Davies, C., Holt, I., Green, J., Harding, J., & Diamond, L. (2009, aug). User needs and implications for modelling vague named places. *Spatial Cognition and Computation*, *9*(3), 174–194. doi: 10.1080/13875860903121830

Deng, J., Dong, W., Socher, R., Li, J., Kai Li, Li Fei-Fei, ... Li Fei-Fei (2009, jun). ImageNet: A large-scale hierarchical image database. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 248–255). IEEE. doi:

10.1109/CVPR.2009.5206848

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2017, jul). Toward Geometric Deep SLAM. , 14.

Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001, jun). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, *17*(3), 229–241. doi: 10.1109/70.938381

Doeller, C. F., Barry, C., & Burgess, N. (2010, feb). Evidence for grid cells in a human memory network. *Nature*, *463*(7281), 657–661. doi: 10.1038/nature08704

Donahue, J., Krähenbühl, P., & Darrell, T. (2016, may). Adversarial Feature Learning. *arXiv preprint arXiv:1605.09782*.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... Brox, T. (2015, dec). Flownet: Learning optical flow with convolutional networks. In *2015 ieee international conference on computer vision (iccv)* (pp. 2758–2766). IEEE. doi: arXiv:1504.06852

Duckham, M., Winter, S., & Robinson, M. (2010, mar). Including landmarks in routing instructions. *Journal of Location Based Services*, *4*(1), 28–52. doi: 10.1080/17489721003785602

Eade, E., & Drummond, T. (2009, apr). Edge landmarks in monocular SLAM. *Image and Vision Computing*, *27*(5), 588–596. doi: 10.1016/j.imavis.2008.04.012

Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2366—-2374). doi: 10.1007/978-3-540-28650-9_5

Elfes, A. (1989, jun). Using occupancy grids for mobile robot perception and navigation. *Computer*, *22*(6), 46–57. doi: 10.1109/2.30720

Eslami, S. M. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., ... Hassabis, D. (2018, jun). Neural scene representation and rendering. *Science*, *360*(6394), 1204–1210. doi: 10.1126/science.aar6170

Fei-Fei, L., & Perona, P. (2005). A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)* (pp. 524–531 vol. 2). doi: 10.1109/CVPR.2005.16

Flynn, J., Neulander, I., Philbin, J., & Snavely, N. (2015). DeepStereo: Learning to Predict New Views from the World's Imagery. In *2015 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 5515–5524). doi: 10.1109/CVPR.2016.595

Gallagher, A., Joshi, D., Jie Yu, & Jiebo Luo. (2009, jun). Geo-location inference from image content and user tags. In *2009 ieee conference on computer vision and pattern recognition workshops (cvprw)* (pp. 55–62). IEEE. doi: 10.1109/CVPR.2009.5204168

Gálvez-López, D., & Tardós, J. D. (2011, sep). Real-time loop detection with bags of binary words. In *Ieee international conference on intelligent robots and systems* (pp. 51–58). IEEE. doi: 10.1109/IROS.2011.6048525

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016, jun). Image Style Transfer Using Con-

volutional Neural Networks. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 2414–2423). IEEE. doi: 10.1109/CVPR.2016.265

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Aiden, E. L., & Fei-Fei, L. (2017, dec). Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States. *Proceedings of the National Academy of Sciences*, *114*(50), 13108–13113. doi: 10.1073/pnas.1700035114

Glocker, B., Izadi, S., Shotton, J., & Criminisi, A. (2013, oct). Real-time RGB-D camera relocalization. In *2013 ieee international symposium on mixed and augmented reality, ismar 2013* (pp. 173–179). IEEE. doi: 10.1109/ISMAR.2013.6671777

Golledge, R. G. (1978, jan). Representing, Interpreting, And Using Cognized Environment. *Papers in Regional Science*, *41*(1), 169–204. doi: 10.1111/j.1435-5597.1978.tb01046.x

Golledge, R. G. (1999). Human wayfinding and cognitive maps. In R. G. Golledge (Ed.), *Wayfinding behavior: Cognitive mapping and other spatial processes* (pp. 5–45). Baltimore and London: The Johns Hopkins University Press. doi: 10.4324/9780203422908

Golledge, R. G., & Stimson, R. J. R. J. (1997). *Spatial behavior : a geographic perspective*. Guilford Press.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). Prentice Hall.

Goodchild, M. F. (2007, nov). Citizens as sensors: the world of volunteered geography. *GeoJournal*, *69*(4), 211–221. doi: 10.1007/s10708-007-9111-y

Goodchild, M. F. (2011). Formalizing Place in Geographic Information Systems. In *Communities, neighborhoods, and health* (pp. 21–33). New York, NY: Springer New York. doi: 10.1007/978-1-4419-7482-2_2

Goodchild, M. F. (2018, may). GIScience for a driverless age. *International Journal of Geographical Information Science*, *32*(5), 849–855. doi: 10.1080/13658816.2018.1440397

Goodfellow, I. (2016, dec). NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (`http://www.deeplearningbook.org`)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2672–2680). Curran Associates, Inc. Retrieved from `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`

Guivant, J. E., Masson, F. R., & Nebot, E. M. (2002, aug). Simultaneous localization and map building using natural features and absolute information. *Robotics and Autonomous Systems*, *40*(2-3), 79–90. doi: 10.1016/S0921-8890(02)00233-6

Guivant, J. E., & Nebot, E. M. (2001, jun). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, *17*(3), 242–257. doi: 10.1109/70.938382

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017, mar). Improved Training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Proceedings of the ieee computer society conference on computer vision and pattern recognition (cvpr 2006)* (Vol. 2, pp. 1735–1742). IEEE. doi: 10.1109/CVPR.2006.100

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005, jun). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*(7052), 801–806. doi: 10.1038/nature03721

Hansen, P., & Browning, B. (2014, sep). Visual place recognition using HMM sequence matching. In *2014 ieee/rsj international conference on intelligent robots and systems* (pp. 4549–4555). IEEE. doi: 10.1109/IROS.2014.6943207

Hartley, R., & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision* (2nd ed.). Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511811685

Hartley, T., Burgess, N., Lever, C., Cacucci, F., & O'Keefe, J. (2000, jan). Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus*, *10*(4), 369–379. doi: 10.1002/1098-1063(2000)10:4⟨369::AID-HIPO3⟩3.0.CO;2-0

Hartley, T., Lever, C., Burgess, N., & O'Keefe, J. (2013, feb). Space in the brain: how the hippocampal formation supports spatial cognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *369*(1635), 20120510–20120510. doi: 10.1098/rstb.2012.0510

Hays, J., & Efros, A. A. (2008, jun). IM2GPS: estimating geographic information from a single image. In *2008 ieee conference on computer vision and pattern recognition* (pp. 1–8). IEEE. doi: 10.1109/CVPR.2008.4587784

Hays, J., & Efros, A. A. (2015). Large-scale Image Geolocalization. In *Multimodal location estimation of videos and images* (pp. 41–62). Cham: Springer International Publishing. doi: 10.1007/978-3-319-09861-6_3

He, K., Zhang, X., Ren, S., & Sun, J. (2016, jun). Deep Residual Learning for Image Recognition. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 770–778). IEEE. doi: 10.1109/CVPR.2016.90

He, X., Zhou, Y., Zhou, Z., Bai, S., & Bai, X. (2018). Triplet-Center Loss for Multi-View 3D Object Retrieval. *arXiv preprint arXiv:1803.06189*. doi: 10.1017/S0953756204001273

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley. doi: 10.2307/1418888

Heinly, J., Schonberger, J. L., Dunn, E., & Frahm, J.-M. (2015, jun). Reconstructing the world* in six days. In *2015 ieee conference on computer vision and pattern recognition (cvpr)* (Vol. 07-12-June, pp. 3287–3295). IEEE. doi: 10.1109/CVPR.2015.7298949

Hermans, A., Beyer, L., & Leibe, B. (2017, mar). In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Horn, B. K., & Schunck, B. G. (1981, aug). Determining optical flow. *Artificial Intelligence*, *17*(1-3), 185–203. doi: 10.1016/0004-3702(81)90024-2

Huang, H. (2016, oct). Context-Aware Location Recommendation Using Geotagged Photos

in Social Media. *ISPRS International Journal of Geo-Information*, *5*(12), 195. doi: 10.3390/ijgi5110195

Huang, H., Gartner, G., Krisp, J. M., Raubal, M., & Van de Weghe, N. (2018, apr). Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, *12*(2), 63–93. doi: 10.1080/17489725.2018.1508763

Hunter, T., Abbeel, P., & Bayen, A. (2014, apr). The path inference filter: Model-based low-latency map matching of probe vehicle data. *IEEE Transactions on Intelligent Transportation Systems*, *15*(2), 507–529. doi: 10.1109/TITS.2013.2282352

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd international conference on international conference on machine learning - volume 37* (Vol. 37, pp. 448–456). JMLR.org. doi: 10.1007/s13398-014-0173-7.2

Irschara, A., Zach, C., Frahm, J.-M., & Bischof, H. (2009, jun). From structure-from-motion point clouds to fast location recognition. In *2009 ieee conference on computer vision and pattern recognition workshop (cvprw)* (pp. 2599–2606). IEEE. doi: 10.1109/CVPRW.2009.5206587

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., & Research, B. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. *arXiv preprint. arXiv:1611.07004*.

Jain, V., & Seung, S. (2009). Natural Image Denoising with Convolutional Networks. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. 21, pp. 769–776). Curran Associates, Inc.

Ji, R., Gao, Y., Liu, W., Xie, X., Tian, Q., & Li, X. (2015, mar). When Location Meets Social Multimedia: A Survey on Vision-based Recognition and Mining for Geo-Social Multimedia Analytics. *ACM Transactions on Intelligent Systems and Technology*, *6*(1), 1–18. doi: 10.1145/2597181

Johns, E., & Yang, G.-Z. (2011, nov). From images to scenes: Compressing an image cluster into a single scene model for place recognition. In *2011 international conference on computer vision (iccv)* (pp. 874–881). IEEE. doi: 10.1109/ICCV.2011.6126328

Jones, C. B., Alani, H., & Tudhope, D. (2001). Geographical Information Retrieval with Ontologies of Place. In D. R. Montello (Ed.), *Spatial information theory. cosit 2001. lecture notes in computer science, vol 2205.* (pp. 323–335). Springer, Berlin, Heidelberg. doi: 10.1007/3-540-45424-1

Kabachnik, P. (2012, mar). Nomads and mobile places: disentangling place, space and mobility. *Identities*, *19*(2), 210–228. doi: 10.1080/1070289X.2012.672855

Kaplan, S. (1973). Cognitive maps in perception and thought. In R. M. Downs & D. Stea (Eds.), *Image and environment: Cognitive mapping and spatial behaviour* (pp. 63–78). Aldine Chicago.

Karaletsos, T., Belongie, S., & Rätsch, G. (2015, jun). Bayesian representation learning with oracle constraints. *arXiv preprint arXiv:1506.05011*. doi: 10.1051/0004-6361/201527329

Kendall, A., & Cipolla, R. (2016, may). Modelling uncertainty in deep learning for camera relocalization. In *2016 ieee international conference on robotics and automation (icra)*

(pp. 4762–4769). IEEE. doi: 10.1109/ICRA.2016.7487679

Kendall, A., & Cipolla, R. (2017). Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 5974–5983). doi: 10.1109/CVPR.2017.694

Kendall, A., Grimes, M., & Cipolla, R. (2016, may). PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *2015 ieee international conference on computer vision (iccv)* (pp. 2938–2946). doi: 10.1109/ICCV.2015.336

Kingma, D. P., & Welling, M. (2013, dec). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kitchin, R. M. (1994, mar). Cognitive maps: What are they and why study them? *Journal of Environmental Psychology*, *14*(1), 1–19. doi: 10.1016/S0272-4944(05)80194-X

Košecká, J., & Zhang, W. (2002). Video Compass. In A. Heyden et al. (Eds.), *Computer vision — eccv 2002* (pp. 476–490). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-47979-1_32

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Lake Tahoe: Curran Associates, Inc.

Kuipers, B. (1978, apr). Modeling spatial knowledge. *Cognitive Science: A Multidisciplinary Journal*, *2*(2), 129–153. doi: 10.1207/s15516709cog0202_3

Laskar, Z., Melekhov, I., Kalia, S., & Kannala, J. (2017, oct). Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. In *2017 ieee international conference on computer vision workshops (iccvw)* (pp. 920–929). IEEE. doi: 10.1109/ICCVW.2017.113

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015, may). Deep learning. *Nature*, *521*(7553), 436–444. doi: 10.1038/nature14539

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., ... Shi, W. (2016, sep). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv preprint arXiv:1609.04802*.

Leonard, J. J., & Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings iros '91:ieee/rsj international workshop on intelligent robots and systems* (pp. 1442–1447). IEEE. doi: 10.1109/IROS.1991.174711

Lever, C., Wills, T., Cacucci, F., Burgess, N. N., & O'Keefe, J. (2002, mar). Long-term plasticity in hippocampal place-cell representation of environmental geometry. *Nature*, *416*(6876), 90–94. doi: 10.1038/416090a

Levinson, J., & Thrun, S. (2010, may). Robust vehicle localization in urban environments using probabilistic maps. In *2010 ieee international conference on robotics and automation* (pp. 4372–4378). IEEE. doi: 10.1109/ROBOT.2010.5509700

Li, F., & Košecká, J. (2006). Probabilistic location recognition using reduced feature set. In *Proceedings - ieee international conference on robotics and automation* (Vol. 2006, pp. 3405–3410). IEEE. doi: 10.1109/ROBOT.2006.1642222

Li, L., & Goodchild, M. F. (2012). Constructing places from spatial footprints. In *Pro-*

ceedings of the 1st acm sigspatial international workshop on crowdsourced and volunteered geographic information - geocrowd '12 (p. 15). New York: ACM Press. doi: 10.1145/2442952.2442956

Li, Y., Crandall, D. J., & Huttenlocher, D. P. (2009, sep). Landmark classification in large-scale image collections. In *2009 ieee 12th international conference on computer vision* (pp. 1957–1964). IEEE. doi: 10.1109/ICCV.2009.5459432

Li, Y., Huang, Q., Kerber, M., Zhang, L., & Guibas, L. (2013). Large-scale joint map matching of GPS traces. In *Proceedings of the 21st acm sigspatial international conference on advances in geographic information systems - sigspatial'13* (pp. 214–223). New York, New York, USA: ACM Press. doi: 10.1145/2525314.2525333

Li, Y., Snavely, N., & Huttenlocher, D. P. (2010). Location recognition using prioritized feature matching. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision – eccv 2010* (pp. 791–804). Berlin, Heidelberg: Springer Berlin Heidelberg.

Li, Y., Snavely, N., Huttenlocher, D. P., & Fua, P. (2016). Worldwide Pose Estimation Using 3D Point Clouds. In A. R. Zamir et al. (Eds.), *Large-scale visual geo-localization* (pp. 147–163). Cham: Springer International Publishing. doi: 10.1007/978-3-319-25781-5_8

Lin, T.-Y., Belongie, S., & Hays, J. (2013, jun). Cross-View Image Geolocalization. In *2013 ieee conference on computer vision and pattern recognition* (pp. 891–898). IEEE. doi: 10.1109/CVPR.2013.120

Lin, T.-Y., Yin Cui, Belongie, S., & Hays, J. (2015, jun). Learning deep representations for ground-to-aerial geolocalization. In *2015 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 5007–5015). IEEE. doi: 10.1109/CVPR.2015.7299135

Liu, L., Zhou, B., Zhao, J., & Ryan, B. D. (2016, dec). C-IMAGE: city cognitive mapping through geo-tagged photos. *GeoJournal*, *81*(6), 817–861. doi: 10.1007/s10708-016-9739-6

Liu, Y., Emery, R., Chakrabarti, D., Burgard, W., & Thrun, S. (2001). Using EM to Learn 3D Models of Indoor Environments with Mobile Robots. In *Proceedings of the eighteenth international conference on machine learning* (p. 643). Morgan Kaufmann Publishers.

Long, J., Shelhamer, E., & Darrell, T. (2015, jun). Fully convolutional networks for semantic segmentation. In *2015 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 3431–3440). IEEE. doi: 10.1109/CVPR.2015.7298965

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh ieee international conference on computer vision* (pp. 1150–1157 vol.2). IEEE. doi: 10.1109/ICCV.1999.790410

Lowe, D. G. (2004, nov). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Lowry, S., Sunderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., & Milford, M. J. (2016, feb). Visual Place Recognition: A Survey. *IEEE Transactions on Robotics*, *32*(1), 1–19. doi: 10.1109/TRO.2015.2496823

Lynch, K. (1960). *The image of the city*. MIT press.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015, nov). Adversarial Autoencoders. *arXiv preprint arXiv:1511.05644*.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). Least Squares Generative Adversarial Networks. In *Ieee international conference on computer vision (iccv 2017).*

McCulloch, W. S., & Pitts, W. (1943, dec). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. doi: 10.1007/BF02478259

Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017a, oct). Image-Based Localization Using Hourglass Networks. In *2017 ieee international conference on computer vision workshops (iccvw)* (pp. 870–877). IEEE. doi: 10.1109/ICCVW.2017.107

Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017b, sep). Relative Camera Pose Estimation Using Convolutional Neural Networks. In J. Blanc-Talon, R. Penne, W. Philips, D. Popescu, & P. Scheunders (Eds.), *Advanced concepts for intelligent vision systems. acivs 2017* (pp. 675–687). Springer, Cham. doi: 10.1007/978-3-319-70353-4_57

Mescheder, L., Nowozin, S., & Geiger, A. (2017, jan). Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. *arXiv preprint arXiv:1701.04722*.

Metz, L., Poole, B., Pfau, D., & Sohl-Dickstein, J. (2016, nov). Unrolled Generative Adversarial Networks. *arXiv preprint arXiv:1611.02163*.

Milford, M. J., & Wyeth, G. F. (2012, may). SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings - ieee international conference on robotics and automation* (pp. 1643–1649). IEEE. doi: 10.1109/ICRA.2012.6224623

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Montello, D. R. (2005). Navigation. In P. Shah & A. Miyake (Eds.), *The cambridge handbook of visuospatial thinking* (pp. 257–294). Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511610448.008

Moravec, H. (1996). *Robot spatial perceptionby stereoscopic vision and 3d evidence grids* (Tech. Rep. No. September). CMU. doi: CMU-RI-TR-96-34

Moravec, H., & Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings. 1985 ieee international conference on robotics and automation* (Vol. 2, pp. 116–121). Institute of Electrical and Electronics Engineers. doi: 10.1109/ROBOT.1985.1087316

Moser, E. I., Kropff, E., & Moser, M.-B. (2008, jun). Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience*, *31*(1), 69–89. doi: 10.1146/annurev.neuro.31.061307.090723

Moser, E. I., Moser, M.-B., & McNaughton, B. L. (2017, oct). Spatial representation in the hippocampal formation: a history. *Nature Neuroscience*, *20*(11), 1448–1464. doi: 10.1038/nn.4653

Mou, L., & Zhu, X. X. (2018, feb). IM2HEIGHT: Height Estimation from Single Monocular Imagery via Fully Residual Convolutional-Deconvolutional Network.

Murillo, A. C., Singh, G., Kosecká, J., & Guerrero, J. J. (2013, feb). Localization in Urban Environments Using a Panoramic Gist Descriptor. *IEEE Transactions on Robotics*, *29*(1), 146–160. doi: 10.1109/TRO.2012.2220211

Newman, P., & Kin Ho. (2005). SLAM-Loop Closing with Visually Salient Features. In *Proceedings of the 2005 ieee international conference on robotics and automation* (Vol. 2005, pp. 635–642). IEEE. doi: 10.1109/ROBOT.2005.1570189

Nicosevici, T., & Garcia, R. (2012, aug). Automatic Visual Bag-of-Words for Online Robot Navigation and Mapping. *IEEE Transactions on Robotics*, *28*(4), 886–898. doi: 10.1109/TRO.2012.2192013

Noh, H., Araujo, A., Sim, J., & Han, B. (2016). Image Retrieval with Deep Local Features and Attention-based Keypoints. , *abs/1612.0*.

O'Keefe, J. (1976, jan). Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, *51*(1), 78–109. doi: 10.1016/0014-4886(76)90055-8

O'Keefe, J., & Burgess, N. (1996, may). Geometric determinants of the place fields of hippocampal neurons. *Nature*, *381*(6581), 425–428. doi: 10.1038/381425a0

O'Keefe, J., & Conway, D. H. (1978, apr). Hippocampal place units in the freely moving rat: why they fire where they fire. *Experimental Brain Research*, *31*(4), 573–590. doi: 10.1007/BF00239813

O'Keefe, J., & Dostrovsky, J. (1971, nov). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, *34*(1), 171–175. doi: 10.1016/0006-8993(71)90358-1

O'Keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map*. Clarendon Press.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3), 145–175. doi: 10.1023/A:1011139631724

Pandey, G., & Dukkipati, A. (2017, may). Variational methods for conditional multimodal deep learning. In *Proceedings of the 2017 international joint conference on neural networks* (Vol. 2017-May, pp. 308–315). IEEE. doi: 10.1109/IJCNN.2017.7965870

Pepperell, E., Corke, P. I., & Milford, M. J. (2014, may). All-environment visual place recognition with SMART. In *2014 ieee international conference on robotics and automation (icra)* (pp. 1612–1618). IEEE. doi: 10.1109/ICRA.2014.6907067

Radford, A., Metz, L., & Chintala, S. (2015, nov). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*.

Radwan, N., Valada, A., & Burgard, W. (2018, apr). VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry.

Raper, J., Gartner, G., Karimi, H., & Rizos, C. (2007a, jun). Applications of locationbased services: a selected review. *Journal of Location Based Services*, *1*(2), 89–111. doi: 10.1080/17489720701862184

Raper, J., Gartner, G., Karimi, H., & Rizos, C. (2007b, mar). A critical evaluation of location based services and their potential. *Journal of Location Based Services*, *1*(1), 5–45. doi: 10.1080/17489720701584069

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016, may). Generative

Adversarial Text to Image Synthesis. *arXiv preprint arXiv:1605.05396*.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. doi: 10.1037/h0042519

Rosten, E., & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer vision eccv 2006.* (Vol. 3951, pp. 430–443). Springer, Berlin, Heidelberg. doi: 10.1007/11744023_34

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, nov). ORB: An efficient alternative to SIFT or SURF. In *2011 international conference on computer vision* (pp. 2564–2571). IEEE. doi: 10.1109/ICCV.2011.6126544

Ruder, S. (2016, sep). An overview of gradient descent optimization algorithms.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986, oct). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. doi: 10.1038/323533a0

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015, dec). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. doi: 10.1007/s11263-015-0816-y

Salesses, P., Schechtner, K., & Hidalgo, C. A. (2013, jul). The collaborative image of the city: mapping the inequality of urban perception. *PLoS ONE*, *8*(7), e68400. doi: 10.1371/journal.pone.0068400

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016, jun). Improved Techniques for Training GANs. *arXiv preprint arXiv:1606.03498*.

Sattler, T., Havlena, M., Radenovic, F., Schindler, K., & Pollefeys, M. (2015, dec). Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In *2015 ieee international conference on computer vision (iccv)* (Vol. 2015 Inter, pp. 2102–2110). IEEE. doi: 10.1109/ICCV.2015.243

Sattler, T., Leibe, B., & Kobbelt, L. (2011, nov). Fast image-based localization using direct 2D-to-3D matching. In *2011 international conference on computer vision* (pp. 667–674). IEEE. doi: 10.1109/ICCV.2011.6126302

Sattler, T., Leibe, B., & Kobbelt, L. (2012). Improving Image-Based Localization by Active Correspondence Search. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer vision – eccv 2012* (pp. 752–765). Berlin, Heidelberg: Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-33718-5_54

Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., & Pajdla, T. (2017, jul). Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (Vol. 2017-Janua, pp. 6175–6184). IEEE. doi: 10.1109/CVPR.2017.654

Sattler, T., Weyand, T., Leibe, B., & Kobbelt, L. (2012). Image Retrieval for Image-Based Localization Revisited. In *Procedings of the british machine vision conference 2012* (pp. 76.1–76.12). British Machine Vision Association. doi: 10.5244/C.26.76

Saurer, O., Baatz, G., Köser, K., Ladický, L., & Pollefeys, M. (2016, feb). Image Based Geo-localization in the Alps. *International Journal of Computer Vision*, *116*(3), 213–225. doi: 10.1007/s11263-015-0830-0

Schölkopf, B., & Mallot, H. A. (1995, jan). View-Based Cognitive Mapping and Path

Planning. *Adaptive Behavior*, *3*(3), 311–348. doi: 10.1177/105971239500300303

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. In *The ieee conference on computer vision and pattern recognition (cvpr 2015)* (pp. 815–823). doi: 10.1109/CVPR.2015.7298682

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., ... Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In *Proceedings of the ieee conference on computer vision and pattern recognition.*

Siegel, A. W., & White, S. H. (1975, jan). The Development of Spatial Representations of Large-Scale Environments. *Advances in Child Development and Behavior*, *10*, 9–55. doi: 10.1016/S0065-2407(08)60007-5

Simonyan, K., & Zisserman, A. (2014, sep). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

Sivic, J., & Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Toward category-level object recognition* (pp. 1470–1477). IEEE. doi: 10.1109/ICCV.2003.1238663

Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3D. In *Proceeding siggraph '06 acm siggraph 2006 papers* (Vol. 25, p. 835). New York, New York, USA: ACM Press. doi: 10.1145/1179352.1141964

Snavely, N., Seitz, S. M., & Szeliski, R. (2008, nov). Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, *80*(2), 189–210. doi: 10.1007/s11263-007-0107-3

Sohn, K., Lee, H., & Yan, X. (2015). Learning Structured Output Representation using Deep Conditional Generative Models. *Advances in Neural Information Processing Systems (NIPS 2015)*, 3483–3491. doi: 10.1007/BF03255766

Sorrows, M. E., & Hirtle, S. C. (1999). The nature of landmarks for real and electronic spaces. In C. Freska & D. M. Mark (Eds.), *Spatial information theory. cognitive and computational foundations of geographic information science. cosit 1999.* (Vol. 1661, pp. 37–50). Berlin, Heidelberg: Springer, Berlin, Heidelberg. doi: 10.1007/3-540-48384-5_3

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014, dec). Striving for Simplicity: The All Convolutional Net.
doi: 10.1163/_q3_SIM_00374

Stumm, E., Mei, C., & Lacroix, S. (2013, nov). Probabilistic place recognition with covisibility maps. In *2013 ieee/rsj international conference on intelligent robots and systems* (pp. 4158–4163). IEEE. doi: 10.1109/IROS.2013.6696952

Sünderhauf, N., & Protzel, P. (2011, sep). BRIEF-Gist - closing the loop by simple means. In *2011 ieee/rsj international conference on intelligent robots and systems* (pp. 1234–1241). IEEE. doi: 10.1109/IROS.2011.6094921

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., & Milford, M. (2015, sep). On the performance of ConvNet features for place recognition. In *2015 ieee/rsj international conference on intelligent robots and systems (iros)* (Vol. 2015-Decem, pp. 4297–4304). IEEE. doi: 10.1109/IROS.2015.7353986

Sünderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., & Milford, M. (2015). Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free. In *Robotics: Science and systems xi.* doi: 10.15607/RSS.2015.XI.022

Surmann, H., Nüchter, A., & Hertzberg, J. (2003, dec). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, *45*(3-4), 181–198. doi: 10.1016/J.ROBOT.2003.09.004

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning : an introduction.* MIT Press.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016, feb). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv preprint arXiv:1602.07261*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016, jun). Rethinking the Inception Architecture for Computer Vision. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 2818–2826). IEEE. doi: 10.1109/CVPR.2016.308

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015, jun). Going deeper with convolutions. In *2015 ieee conference on computer vision and pattern recognition (cvpr)* (Vol. 07-12-June, pp. 1–9). IEEE. doi: 10.1109/CVPR.2015.7298594

Szeliski, R., & Kang, S. B. (1994, mar). Recovering 3d shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, *5*(1), 10–28. doi: 10.1006/jvci.1994.1002

Tadmor, O., Wexler, Y., Rosenwein, T., Shalev-Shwartz, S., & Shashua, A. (2016, may). Learning a metric embedding for face recognition using the multibatch method. *arXiv preprint arXiv:1605.07270*.

Thorndyke, P. W., & Goldin, S. E. (1983). Spatial Learning and Reasoning Skill. In H. L. J. Pick & L. P. Acredolo (Eds.), *Spatial orientation* (pp. 195–217). Boston, MA: Springer US. doi: 10.1007/978-1-4615-9325-6_9

Thrun, S. (2003). Robotic mapping: a survey. In G. Lakemeyer & B. Nebel (Eds.), *Exploring artificial intelligence in the new millennium* (pp. 1–35). Morgan Kaufmann Publishers.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, *55*(4), 189–208. doi: 10.1037/h0061626

Tuan, Y.-f. (1977). *Space and place : the perspective of experience.* University of Minnesota Press.

Tzeng, E., Zhai, A., Clements, M., Townshend, R., & Zakhor, A. (2013, jun). User-Driven Geolocation of Untagged Desert Imagery Using Digital Elevation Models. In *2013 ieee conference on computer vision and pattern recognition workshops* (pp. 237–244). IEEE. doi: 10.1109/CVPRW.2013.42

Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., & Brox, T. (2017, jul). DeMoN: Depth and Motion Network for Learning Monocular Stereo. In *2017*

ieee conference on computer vision and pattern recognition (cvpr) (pp. 5622–5631). IEEE. doi: 10.1109/CVPR.2017.596

Valada, A., Radwan, N., & Burgard, W. (2018, mar). Deep Auxiliary Learning for Visual Localization and Odometry.

van der Maaten, L., & Hinton, G. E. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Vasardani, M., Winter, S., & Richter, K. F. (2013, dec). Locating place names from place descriptions. *International Journal of Geographical Information Science*, *27*(12), 2509–2532. doi: 10.1080/13658816.2013.785550

Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., & Fragkiadaki, K. (2017, apr). SfM-Net: Learning of Structure and Motion from Video.

Vo, N., Jacobs, N., & Hays, J. (2017, oct). Revisiting IM2GPS in the Deep Learning Era. In *2017 ieee international conference on computer vision (iccv)* (Vol. 2017-Octob, pp. 2640–2649). IEEE. doi: 10.1109/ICCV.2017.286

Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., & Cremers, D. (2017, oct). Image-Based Localization Using LSTMs for Structured Feature Correlation. In *2017 ieee international conference on computer vision (iccv)* (Vol. 2017, pp. 627–637). IEEE. doi: 10.1109/ICCV.2017.75

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., ... Wu, Y. (2014, jun). Learning fine-grained image similarity with deep ranking. In *Proceedings of the ieee computer society conference on computer vision and pattern recognition (cvpr 2014)* (pp. 1386–1393). IEEE. doi: 10.1109/CVPR.2014.180

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *European conference on computer vision (eccv 2016)* (Vol. 9911 LNCS, pp. 499–515). Springer, Cham. doi: 10.1007/978-3-319-46478-7_31

Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In *System modeling and optimization* (pp. 762–770). Berlin/Heidelberg: Springer-Verlag. doi: 10.1007/BFb0006203

Werner, S., Krieg-Brückner, B., & Herrmann, T. (2000). Modelling Navigational Knowledge by Route Graphs. In *Spatial cognition ii, integrating abstract theories, empirical studies, formal methods, and practical applications* (pp. 295–316). Springer. doi: 10.1007/3-540-45460-8_22

Weyand, T., Kostrikov, I., & Philbin, J. (2016, oct). PlaNet - Photo Geolocation with Convolutional Neural Networks. In *Computer vision eccv 2016* (pp. 37–55). Springer, Cham. doi: 10.1007/978-3-319-46484-8_3

Wilson, M. A., & McNaughton, B. L. (1993, aug). Dynamics of the hippocampal ensemble code for space. *Science*, *261*(5124), 1055–1058. doi: 10.1126/science.8351520

Winter, S., & Freksa, C. (2012, dec). Approaching the notion of place by contrast. *Journal of Spatial Information Science*, *no. 5*(5), 31–50. doi: 10.5311/JOSIS.2012.5.90

Wu, J., Ma, L., & Hu, X. (n.d., may). Delving deeper into convolutional neural networks for camera relocalization. In *2017 ieee international conference on robotics and automation (icra)* (pp. 5644–5651). IEEE. doi: 10.1109/ICRA.2017.7989663

Yin, Z., & Shi, J. (2018, mar). GeoNet: Unsupervised Learning of Dense Depth, Optical

Flow and Camera Pose.

Zamir, A. R., & Shah, M. (2010). Accurate image localization based on google maps street view. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision – eccv 2010* (pp. 255–268). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zamir, A. R., & Shah, M. (2014, aug). Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(8), 1546–1558. doi: 10.1109/TPAMI.2014.2299799

Zamir, A. R., Wekel, T., Agrawal, P., Wei, C., Malik, J., & Savarese, S. (2016). Generic 3D representation via pose estimation and matching. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *European conference on computer vision* (Vol. 9907 LNCS, pp. 535–553). Amsterdam. doi: 10.1007/978-3-319-46487-9_33

Zhang, F., Zhou, B., Liu, L., Liu, Y., Fung, H. H., Lin, H., & Ratti, C. (2018, dec). Measuring human perceptions of a large-scale urban region using machine learning. *Landscape and Urban Planning*, *180*, 148–160. doi: 10.1016/j.landurbplan.2018.08.020

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. (2017, dec). StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *Proceedings of the ieee international conference on computer vision* (Vol. 2017-Octob, pp. 5908–5916). IEEE. doi: 10.1109/ICCV.2017.629

Zhang, X., Fang, Z., Wen, Y., Li, Z., & Qiao, Y. (2017, Oct). Range Loss for Deep Face Recognition With Long-Tailed Training Data. In *The ieee international conference on computer vision (iccv).* IEEE.

Zheng, Y.-T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., ... Neven, H. (2009, jun). Tour the world: Building a web-scale landmark recognition engine. In *2009 ieee conference on computer vision and pattern recognition* (pp. 1085–1092). IEEE. doi: 10.1109/CVPRW.2009.5206749

Zhou, B., Liu, L., Oliva, A., & Torralba, A. (2014). Recognizing City Identity via Attribute Analysis of Geo-tagged Images. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision eccv 2014.* (Vol. 8691, pp. 519–534). Springer, Cham. doi: 10.1007/978-3-319-10578-9_34

Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017, jul). Unsupervised Learning of Depth and Ego-Motion from Video. In *2017 ieee conference on computer vision and pattern recognition (cvpr)* (pp. 6612–6619). IEEE. doi: 10.1109/CVPR.2017.700

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., ... Zeeb, E. (2014). Making bertha drive-an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, *6*(2), 8–20. doi: 10.1109/MITS.2014.2306552

Appendix A

KL-divergence Between Two Gaussian Distributions

In this section, we describe how to derive the KL-divergence between two Gaussian distributions. If the two Gaussians are univariate denoted as $p(x) = \mathcal{N}(\mu_1, \sigma_1^2)$ and $q(x) = \mathcal{N}(\mu_2, \sigma_2^2)$, then we have

$$\begin{aligned} D_{\mathrm{KL}}(p, q) &= -\int p(x) \log q(x) dx + \int p(x) \log p(x) \mathrm{d}x \\ &= \frac{1}{2} \log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + \log 2\pi\sigma_1^2) \\ &= \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \end{aligned}$$

If they are two multivariate Gaussians that are denoted as $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2)$ and $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2)$, then we have

$$\begin{aligned} D_{\mathrm{KL}}(p, q) &= \int \left[\log(p(\boldsymbol{x})) - \log(q(\boldsymbol{x}))\right] p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &= \int \left[\frac{1}{2} \log \frac{|\boldsymbol{\sigma}_2^2|}{|\boldsymbol{\sigma}_1^2|} - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\sigma}_1^{2^{-1}} (\boldsymbol{x} - \boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\sigma}_2^{2^{-1}} (\boldsymbol{x} - \boldsymbol{\mu}_2)\right] \times p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2} \log \frac{|\boldsymbol{\sigma}_2^2|}{|\boldsymbol{\sigma}_1^2|} - \frac{1}{2} \mathrm{tr} \, \left\{\mathbf{E}[(\boldsymbol{x} - \boldsymbol{\mu}_1)(\boldsymbol{x} - \boldsymbol{\mu}_1)^T] \, \boldsymbol{\sigma}_1^{2^{-1}}\right\} + \frac{1}{2} \mathbf{E}[(\boldsymbol{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\sigma}_2^{2^{-1}} (\boldsymbol{x} - \boldsymbol{\mu}_2)] \\ &= \frac{1}{2} \log \frac{|\boldsymbol{\sigma}_2^2|}{|\boldsymbol{\sigma}_1^2|} - \frac{1}{2} \mathrm{tr} \, \{I_d\} + \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\sigma}_2^{2^{-1}} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \mathrm{tr}\{\boldsymbol{\sigma}_2^{2^{-1}} \boldsymbol{\sigma}_1^2\} \\ &= \frac{1}{2} \left[\log \frac{|\boldsymbol{\sigma}_2^2|}{|\boldsymbol{\sigma}_1^2|} - d + \mathrm{tr}\{\boldsymbol{\sigma}_2^{2^{-1}} \boldsymbol{\sigma}_1^2\} + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{\sigma}_2^{2^{-1}} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\right]. \end{aligned}$$

Appendix B

The ELBO of Center-Triplet-VAE

In this section, we describe how to derive the evidence lower bound of Center-Triplet-VAE.

$$
\begin{aligned}
&\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \\
&= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \log \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \right] \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \left[\log(\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})}) + \log(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})})(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}, \boldsymbol{c}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})}) \right] \mathrm{d}\boldsymbol{z} \\
&= D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \right) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)})
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \\
&= \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} + \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}, \boldsymbol{c}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})}) \right] \mathrm{d}\boldsymbol{z} \\
&\text{suppose } p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}) \approx p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)}) \\
&= -D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})) + \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}, \boldsymbol{c}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})}) \mathrm{d}\boldsymbol{z} \\
&= -D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})) + \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}|\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \\
&\text{suppose } p(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}|\boldsymbol{z}) \text{ can be factorized as } p(\boldsymbol{x}^{(i)}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{c}^{(i)}|\boldsymbol{z}) \\
&= -D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})) + \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{c}^{(i)}|\boldsymbol{z}) \right] \mathrm{d}\boldsymbol{z} \\
&= -D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}^{(i)})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{c}^{(i)}|\boldsymbol{z})
\end{aligned}
$$

Appendix C

The ELBO of Clustering Center-Triplet-VAE

In this section, we describe how to derive the evidence lower bound of clustering Center-Triplet-VAE for place representation learning.

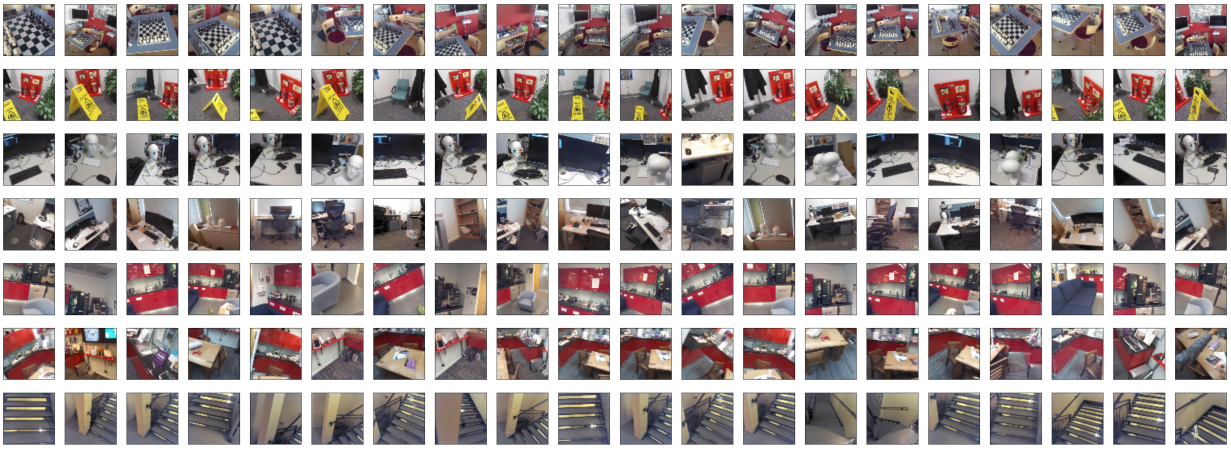
$$
\begin{aligned}
&\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) \\
&= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) \int_{\boldsymbol{z}} \sum_{\boldsymbol{c}} q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} \sum_{\boldsymbol{c}} q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} \sum_{\boldsymbol{c}} q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \log \left[p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})} \right] \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} \sum_{\boldsymbol{c}} q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \left[\log(\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})}) + \log(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}, \boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})} \right] \mathrm{d}\boldsymbol{z} \\
&\geq D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \right) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)})
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = \int_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \left[\log(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}, \boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})} \right] \mathrm{d}\boldsymbol{z} \\
&\text{suppose } q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)}) \text{ can be factorized as } q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c}) q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}^{(i)}) \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{c}|\boldsymbol{x}^{(i)})} \left[\log(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}^{(i)})} + \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{c})} + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}, \boldsymbol{c}) \right]
\end{aligned}
$$

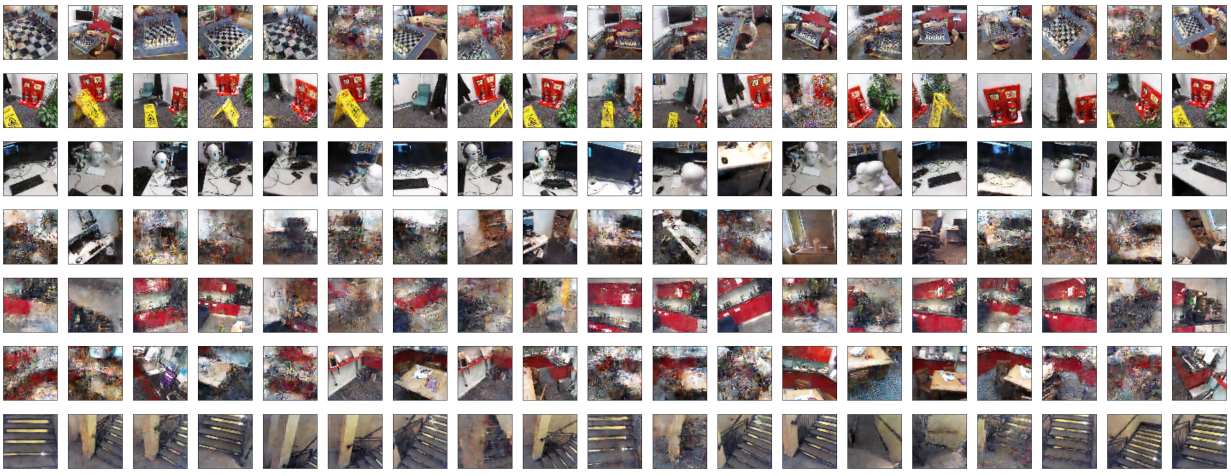
Appendix D

Results of Learning Places Representations with Triplet-VAE

The complete list of test images and their reconstructions is given below (next page).

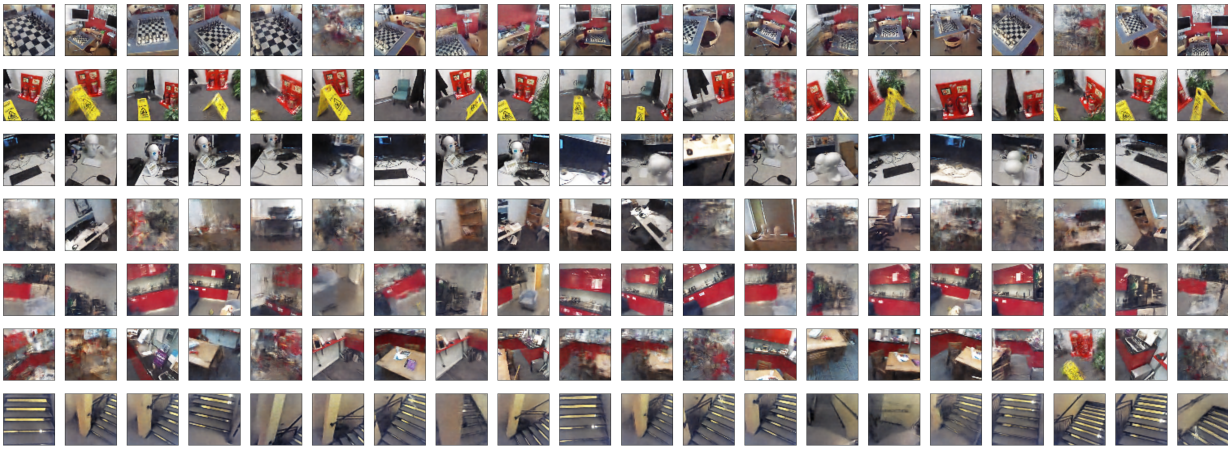


(a)



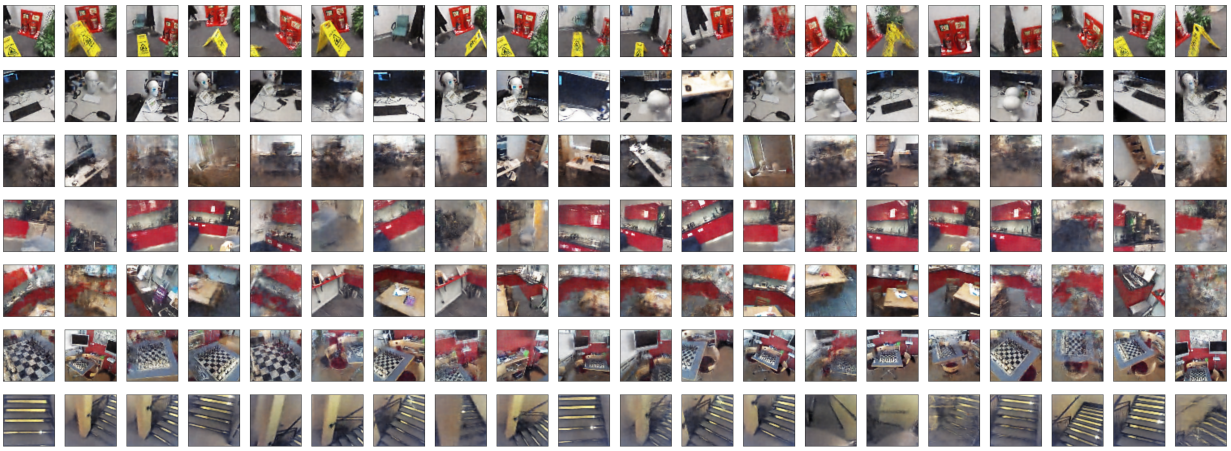
(b)

Figure D.1: The full list of 140 test images and their reconstructions from different methods. (a) original test images, (b) reconstructed by auto-encoder, (c) reconstructed by VAE, (d) reconstructed by Triplet-VAE.



VAE reconstructed test images

(c)



Triplet-VAE reconstructed images

(d)

Figure D.1: The full list of 140 test images and their reconstructions from different methods. (a) original test images, (b) reconstructed by auto-encoder, (c) reconstructed by VAE, (d) reconstructed by Triplet-VAE.

List of Figures

List of Tables

List of Acronym

6DoF	6 degree of freedom
AAE	adversarial autoencoder
Adam	adaptive moment estimation
ANN	artificial neural network
BoW	bag of words
BoVW	bag of visual words
CNN	convolutional neural network
CRF	conditional random field
DCGAN	deep convolutional generative adversarial network
DEM	digital elevation map
GAN	generative adversarial network
GIR	geographical information retrieval
GMM	Gaussian mixture model
GNSS	global navigation satellite system
GPS	global positioning system
HD map	high definition map
HMM	hidden markov model
ILSVRC	ImageNet large scale visual recognition challenge
IMU	inertial measurement unit
KL-divergence	KullbackLeibler divergence
KNN	k-nearest neighbors
LBS	location-based services
LRS	landmark, route, survey model
LSTM	long-short term memory
MAP	maximum a posteriori
MSE	mean squared error
NIPS	neural information processing systems
POI	point of interests
RANSAC	random sample consensus
ReRL	rectified linear unit
RFID	radio-frequency identification
RNN	recurrent neural network
SCL	street centerline

SfM	structure from motion
SGD	stochastic gradient descent
SLAM	simultaneous localization and mapping
SPP	spatial pyramid pooling
SVM	support vector machine
SO group	special orthogonal group
SE group	special Euclidean group
VAE	variational autoencoder
VGI	volunteered geographic information
WGAN	wasserstein generative adverarial network

hand-crafted features in computer vision

SIFT	scale-invariant feature transform
SURF	speed-up robust feature
FAST	features from accelerated segment test
BRIEF	binary robust independent elementary features
ORB	oriented fast and rotated BRIEF
VLAD	vector of locally aggregated descriptions

Acknowledgements

Having finished the thesis writing, I can mark a beautiful ending for my 6-years Ph.D. study in Munich, Germany. First and foremost, I would like to give my sincere thanks to my supervisor Prof. Liqiu Meng. She opened the door to the academic world for me. She provides me an excellent platform and enough patience for my research, so I can enjoy the freedom of conducting my research. She also consistently encouraged me to develop my research interest and to train my soft skills. I appreciate the fascinating stories and life experience she shared with us during coffee breaks.

I am grateful to Prof. Dipl.-Ing. Dr. Wolfgang Kainz for reviewing my thesis and acting as co-supervisor. His valuable comments and insights improve my understanding of the "place modeling" problem, and inspire further development of my place model. Besides my supervisors, Prof. Jukka Krisp guided my research during my first two years and has been continuously paying attention to my research work.

Especially, I am grateful to my wife, Dr. Guiying Du. She consistently encourages and supports my research and thesis writing. She has finished her Ph.D. program within 3 years. She as a role model motivated me and pushed my thesis writing fast forward. In a long period, discussing our research topics consisted of a large portion of our live. I was very much inspired by her research style and problem-solving methods. I'm also proud of having her name in both of my master thesis and doctoral dissertation.

I very much appreciate working at Lehrstuhl fr Kartographie (LfK) of Technische Universitt Mnchen and enjoy being a member of the research group. Thanks to Luise Fleiner, she helped me a lot for dealing with administrative staffs; Holger Kumke helped me built a computation workstation, so I'm able to run my neural network models. Thanks to Linfang Ding for her guidance and help in scientific paper writing. I also appreciate Jian Yang's consistent support and encouragement. I enjoyed playing basketball with him as well. I also appreciate the discussions and collaborations from Mathias Jahnke, Juliane Cron, and Christian Murphy. Being in the same project, Ekaterina Chuprikova always kept me updated with the project and new information from IGSSE. I appreciate Stefan Peters for his kindness and sense of humor which helped me to get accustomed to living in Germany during my first days here. Thanks to Lianhuan Wei, Xiao Xie, and Wanli Lou

for sharing their working and research experience in China. Thanks to all my colleagues of the LfK for always providing a friendly and fun working atmosphere in the group.

The thesis was funded by China Scholarship Council (CSC). My research activities including attending conferences, courses, and skill training are supported by the project "Sense-making image mapping from remotely sensed GLC30m", which was funded by the International Graduate School of Science and Engineering (IGSSE), TUM. The experiments in the thesis have profited from the NVIDIA donation of the Titan Xp graphics card. I gratefully acknowledge all supports. Without them, this thesis would not have been possible.

Finally, I want to thank my mother who has encouraged and supported me without any condition in my way of pursuing my degree. My warm gratitude also goes to all my friends in Munich for making my life interesting and fruitful.