LEARNING ADAPTIVE REPRESENTATIONS FOR IMAGE RETRIEVAL AND RECOGNITION

Hyo Jin Kim

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2018

Approved by:

Jan-Michael Frahm

Alexander C. Berg

Marc Niethammer

Enrique Dunn

Torsten Sattler

ABSTRACT

Hyo Jin Kim: Learning Adaptive Representations for Image Retrieval and Recognition
(Under the direction of Jan-Michael Frahm)

Content-based image retrieval is a core problem in computer vision. It has a wide range of application such as object and place recognition, digital library search, organizing image collections, and 3D reconstruction. However, robust and accurate image retrieval from a large-scale image collection still remains an open problem. For particular instance retrieval, challenges come not only from photometric and geometric changes between the query and the database images, but also from severe visual overlap with irrelevant images. On the other hand, large intra-class variation and inter-class similarity between semantic categories represents a major obstacle in semantic image retrieval and recognition.

This dissertation explores learning image representations that adaptively focus on specific image content to tackle these challenges. For this purpose, three kinds of image contexts for discriminating relevant and irrelevant image content are exploited: (1) local image context, (2) semi-global image context, and (3) global image context. Novel models for learning adaptive image representations based on each context are introduced. Moreover, as a byproduct of training the proposed models, the underlying task-relevant contexts are automatically revealed from the data in a self-supervised manner. These include data-driven notion of *good* local mid-level features, task-relevant semi-global contexts with rich high-level information, and the hierarchy of images. Experimental evaluation illustrates the superiority of the proposed methods in the applications of place recognition, scene categorization, and particular object retrieval.

ACKNOWLEDGEMENTS

I would like to thank Jesus for his grace, guidance, protection, and provisions throughout my life, including my journey to PhD, from the application to the defense, and beyond.

I am blessed to be advised by Prof. Jan-Michael Frahm, who took me under his wing and helped me to develop as a researcher and a person. Without his help and guidance, this work would not have been possible. I deeply appreciate his encouragement and extensive support throughout the years. Especially, I am grateful for his patience and providing me the freedom to explore and develop ideas on my own. I would further like to thank his lovely family for kindly inviting us over for celebrations, and for the fun memories.

I would also like to extend my gratitude to my wonderful committee for their genuine investment on my work as well as their kind support. I had a great pleasure working with Prof. Enrique Dunn, who provided valuable input on this work. His trust and support also meant a lot to me. My sincere thanks goes out to Prof. Alex Berg for fruitful discussions, crucial research advice when I was facing obstacles, and always willing to help me. Many thanks to Prof. Marc Niethammer for being a great teacher, for giving me the opportunity to help teach his class, as well as for his help and encouragement during my tough times. I am deeply grateful to Dr. Torsten Sattler, on whose work I have drawn upon since the beginning of my PhD study, for his insightful and sharp feedback on this work.

In addition, I must thank the researchers who generously provided me with the data and code, as well as the answers to my questions: Dr. Amir Zamir, Dr. Relja Arandjelović, Prof. Akihiko Torii, Dr. Giorgos Tolias, and Dr. Bolei Zhou.

Special thanks to my colleagues who were always there for me, providing me with helpful discussions, proofreading, and the pleasant learning environment: Jared Heinly, Enliang Zheng, Yillin Wang, Yi Xu, Dinghuang Ji, Ke Wang, David Perra, Meng Tan, Sangwoo

Cho, Hongsheng Yang, Johannes Schönberger, Rohit Gupta, True Price, Akash Bapat, Marc Eder, Zhen Wei, John Lim, Joe Tighe, Xufeng Han, Vincente Ordonez, Hadi Kiapour, Wei Liu, Sirion Vittayakorn, Eunbyung Park, Yipin Zhou, Licheng Yu, Chengyang Fu, Philip Ammirato, Patrick Poirson, Misha Shvets, Adam Aji. I would like to especially thank Jared for his help during my first two years.

Furthermore, I would like to thank Prof. Kyoung Mu Lee and Prof. Minsu Cho for their continuous support and encouragement. I am also thankful to Prof. Vladimir Jojic and Prof. Tamara Berg for their help and advice.

Finally, I would like to thank my beloved family who accompanied me on this journey. I cannot begin to express how thankful I am to my father, the original Dr. Kim, for his love and devotion. I am deeply indebted to him for his mentorship throughout my life, listening to me ramble about my day, providing support and guidance both intellectually and emotionally, and being a great role model for me. I would like to extend my sincere thanks to my mother for her unconditional love and care. I am grateful for her support during my PhD, including her visits to Chapel Hill despite the long, tiring trip from Korea. Many thanks to my sister for her love and for being my best friend, always giving me a reason to smile. I cherish the time we spent together when she visited UNC as an exchange student. I would like to further thank my grand parents and relatives for their prayers and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

We are living in an era where billions of digital photos, and half a billion hours of video are uploaded on the web daily (Cakebread, 2017; Nicas, 2017). Finding an image that we are interested in among those massive image collections is a challenging task. Typically, we are only interested in a few of them, while the vast majority are unrelated ones. Most of the current search engines rely on keywords and textual meta-data for searching and organizing images. However, the meta-data is often noisy and lacks description about the content of the image. Obtaining a detailed and accurate text annotation for an image, on the other hand, is expensive and impossible at times.

The objective of this thesis is to create fully-automatic systems for performing large-scale image search and recognition purely based on visual information, using only the content of an image. As the old saying goes: a picture is worth a thousand words—the amount of information that an image can contain is enormous, and it eliminates the need for manual annotations or meta-data. It can also account for visual concepts that are not easily describable by words. The applications of content-based image retrieval includes organizing photo collections (Johnson et al., 2010; Raguram et al., 2011), digital library search (Chen et al., 2011b), photo-based product search (Kiapour et al., 2015; Liu et al., 2016), reconstructing 3d models from images depicting the same place (Frahm et al., 2010; Heinly et al., 2015), and place recognition (Arandjelović et al., 2016; Sattler et al., 2017).

An image retrieval pipeline largely consists of two parts. The first part is computing the visual similarity between the images. In order to endow computers with the ability to evaluate the similarity between images, we need to provide them with visual representations that captures the essence of an image. Such a representation typically takes the form of a

vector, or a set of vectors. By measuring the distance between these vectors, we can compute the visual similarity between the corresponding images. The second part is to make the search efficient in terms of memory and speed. It often involves approximate nearest neighbor search, that includes quantization and indexing. This dissertation mostly focuses on the former, measuring the image similarity, which determines the accuracy of the search. A brief overview of approximate nearest neighbor search can be found in Appendix A.

In particular instance retrieval, the goal is to find the image with the same instance depicted in the query. The challenge lies in developing a representation that is robust to different variations of the relevant images (i.e., change of a scale, illumination, viewpoint, and occlusion), while being discriminative enough to distinguish the relevant images from the vast majority of irrelevant images. Typically, the larger the size of the database, the more likely to encounter irrelevant images that exhibit larger visual overlap with the query than the relevant images. Thus, in the representation space, we need to somehow minimize the variance between the relevant images, and maximize the distance to the irrelevant images, such that the distances to the relevant images are smaller than those to the irrelevant images.

Semantic image retrieval, on the other hand, is the problem of finding images with the same semantic meaning as the query image, which is an image recognition problem. In fact, many search engines use object or scene recognition to parse and encode semantic meaning from images. Image recognition is not at all different from image retrieval in terms of learning image representations. Instead of two classes, relevant and irrelevant, multiple semantic classes are considered. However, the objective is the same. We want the distance between the samples belonging to the same class to be small, and the distance between samples belonging to different classes to be large in the feature space, such that they can be easily distinguished. Moreover, it shares a similar obstacle as particular instance retrieval, which is the large intra-class variation and inter-class similarity. There are large visual variations in the images with the same semantic meaning while large visual overlap exists between the

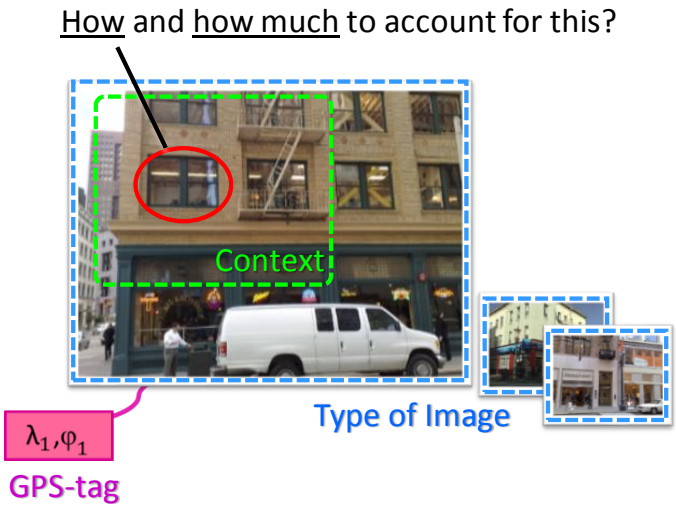


Figure 1.1: How and how much should we account for a visual element when developing an image representation? It may depend on the type of local visual element (local context), its surroundings (semi-global context), and the type of the image itself (global context).

images of different classes. The overlap becomes more severe as the number of semantic classes increases.

In both cases, it is observed that not all of the image content is useful for a given task. Hence, instead of using all image content to represent an image, we would like our visual representations to intelligently emphasize useful regions to promote similarity to the relevant images while suppressing the regions that cause overlap to the irrelevant images. However, based on what can we discriminate these two types of image content? In this thesis, we explore three kinds of image contexts for discriminating a relevant and irrelevant image content: (1) local image context, (2) semi-global image context, and (3) global image context.

First, we exploit local image context for determining task-relevant and -irrelevant image content. For example, a window on a building (Fig. 1.1) is more useful than t-shirts or car wheels for landmark retrieval. Many approaches have been proposed to selectively attend on local regions based on their *uniqueness* (Knopp et al., 2010; Schindler et al., 2007; Arandjelović and Zisserman, 2014a). However, it may be risky to discard all non-unique regions, some of which may contribute to correct retrieval, either by themselves or in combination with others. Instead, we take a data-driven approach to select local regions.

Secondly, we consider a semi-global image context. For example, again in the case of landmark retrieval, windows on buildings are useful, whereas windows on vehicle may introduce obfuscating cues (Fig. 1.1). Therefore, we design our visual representation to adaptively account for a visual element based on its semi-global context. Apart from previous work that uses supervised priors (Mousavian and Košecka, 2015; Torii et al., 2015b), we take advantage of end-to-end learning such that task-relevant context emerges automatically from the data.

Thirdly, we explore the global context of the image. There are certain types of images that are easily confused. For example, the image shown in Fig. 1.1 is less likely to be confused with mountains or beach, than with the images of buildings. Thus, we take global context into account to determine which image group it belongs to. We then develop representations that are specialized for distinguishing images belonging to each particular group. Unlike previous work that organizes classes into coarse categories (Goo et al., 2016; Zhao et al., 2011; Hwang and Sigal, 2014; Deng et al., 2014), we design these groups to disentangle intra-class variation and inter-class similarity at the sub-category level.

We are not the first to consider discriminating relevant and irrelevant image content for image retrieval and recognition. However, most of the existing work relies on supervised priors for determining useful image content. The key difference that set us aside apart from the most of previous work is that we take a data-driven approach. We design our pipelines such that by training our model, the underlying task-relevant contexts are automatically discovered from the data. These include data-driven notion of good local mid-level features, task-relevant semi-global context, and hierarchy of images. Furthermore, we use minimal supervision such as gps-tags. Such a data-driven approach not only allows us to achieve better performance by analyzing the vast amount of data, which is beyond the capability of a human, but also enables our method to generalize to other tasks.

To summarize, this dissertation proposes learning image representations to adaptively focus on specific image content, based on the aforementioned contexts: local image context,

semi-global image context, and global image context. Experimental evaluation demonstrates that the resulting image representation effectively focuses on useful details to promote similarity between relevant images and to deal with visual overlap with irrelevant images in the applications of place recognition, scene categorization, particular object retrieval and object retrieval.

1.1 Thesis Statement

The robustness of image retrieval and recognition can be improved by learning image representations that adaptively focus on specific image content based on (1) local context, (2) semi-global context, and (3) global context.

1.2 Contributions

This dissertation contains several significant contributions that advanced the state of the art in image retrieval and recognition. These contributions include:

Data-driven feature selection and a novel feature description: We propose to discover features that are useful for recognizing a place in a data-driven manner, and use this knowledge to predict useful features in a query image prior to the geo-localization process. This allows achieving better performance while reducing the number of features. Also, for both learning to predict features and retrieving geo-tagged images from the database, we propose a per-bundle vector of locally aggregated descriptors (PBVLAD), where each maximally stable region is described by a vector of locally aggregated descriptors (VLAD) on multiple scale-invariant features detected within the region. Experimental results show the proposed approach achieves a significant improvement over the baselines.

Context-aware feature reweighting: We propose a novel model for learning image representations that integrate context-aware feature reweighting in order to effectively focus

on regions that positively contribute to geo-localization. In particular, we introduce a Contextual Reweighting Network (CRN) that predicts the importance of each region in the feature map based on the image context. This model is learned end-to-end for the image geo-localization task, and requires no annotation other than image geo-tags for training. In experimental results, the proposed approach significantly outperforms the previous state-of-the-art on the standard geo-localization benchmark datasets. We also demonstrate that our CRN discovers task-relevant contexts without any additional supervision.

Disentangling intra-class and inter-class variation in scene categories: We propose a hierarchical generalist-specialist model that automatically builds itself based on the unsupervised discovery of confusing clusters in a coarse to fine manner. The confusing clusters allow specialists to focus on subtle differences between images that are visually similar and confusable to their parents. We also propose a novel alternating architecture that effectively takes advantage of two complementary representations, which captures spatial layouts and transient objects. Experimental results demonstrate that our method significantly outperforms the baselines including the tree-structured models based on coarse categories. As additional innovations, we introduce a novel routing function as well as mini-batch soft k-means for end-to-end fine tuning. Beyond the detailed innovations, our proposed algorithm is generalizable to other categorization tasks, and is applicable to any CNN architecture.

1.3 Thesis Outline

- **Chapter 3. Predicting Good Features for Image Geo-Localization**

This chapter describes our adaptive image representation based on local image context (Kim et al., 2015). The place recognition problem is considered, where the goal is to recognize a place depicted in a query image by using a large database of geo-tagged images at a city-scale.

We take an instance retrieval approach to tackle this problem by finding an image that depicts the same place as the query in the database. We explore the data-driven notion of *good* local features for place recognition. The goal is to foster features having relatively high matching scores in correct localization outcomes, in contrast to their relatively low score for negative outcomes. The feature score prediction is cast a classification problem, and we generate training data automatically from a separate set of geo-tagged Internet images. To cope with noise and high intra-class variation among the training data, we adopt bottom-up clustering techniques. As for our mid-level feature, we propose a per-bundle vector of locally aggregated descriptors (PBVLAD) as a novel representation for a bundled feature that is effective for both learning to predict features and image retrieval. At the query phase, the algorithm selects features in a query image prior to the geo-localization process by accumulating predictions from a bank of linear SVMs. Our results show improved performance is achieved by using only features that are predicted as useful, while reducing the number of features significantly.

- **Chapter 4. Learned Contextual Feature Reweighting for Image Geo-Localization**

This chapter outlines a novel model for learning image representations that integrates feature reweighting based on semi-global image context (Kim et al., 2017a). We propose a novel end-to-end convolutional neural network (CNN) model for learning image representations that adaptively reweight features based on the image context. In particular, we introduce a *Contextual Reweighting Network* (CRN), an auxiliary network that can be used with any standard deep architecture. This chapter also describes constructing the datasets and the details of training the model. We also show how task-relevant contexts are discovered in an unsupervised manner, even though no ground-truth for the weighting nor the context information is provided. Visualizations of learned contexts illustrate that the discovered contextual information contains rich high-level information that is not restricted to semantic cues, such as different types of buildings, vehicles, vegetation, and ground, but includes

structural cues like lattice structure, different perspectives of buildings, and architectural styles themselves.

- **Chapter 5. Hierarchy of Alternating Specialists for Scene Recognition**

This chapter discusses representation selection based on global image context. One of the major challenges in large-scale scene classification is large intra-class variation and inter-class similarity. We propose to group images based on high level features rather than their class membership and dedicate a specialist model per group. In this chapter, we describe a hierarchy of specialists that is learned end-to-end for recognizing scene categories while performing clustering in the learned feature space to discover the hierarchy in an unsupervised manner. The proposed method utilizes two complementary architectures—a global ordered- and an orderless- representation—to account for both coarse layout of the scene and transient objects. A mini-batch soft k-means is also introduced, which allows end-to-end training for fine-tuning. As a byproduct of training our hierarchical model, the hierarchy of scenes, as well as visual concepts that split one specialty area from the other is learned in a data-driven manner.

CHAPTER 2: BACKGROUND AND RELATED WORKS

In this chapter, we first introduce related work in the areas of content-based image retrieval (Sec. 2.1) and visual place recognition (Sec. 2.2). We then discuss several works on automatic training data generation for image retrieval and visual place recognition (Sec. 2.3), tree-structured deep neural networks (Sec. 2.4), and scene category recognition (Sec. 2.5), that are most closely related to our work.

2.1 Content-Based Image Retrieval

Content-based image retrieval can be summarized as finding the most relevant images from the database, using the content of the given query image. It can be largely divided into particular instance retrieval and semantic image retrieval which are described in the following.

2.1.1 Particular Instance Retrieval

Particular instance retrieval is the problem of retrieving images that contain the exact same instance as depicted in the query image. For example, retrieving a specific building, such as *Radcliffe Camera*, or retrieving a specific object, such as the CD cover of *Abbey Road*. The image retrieval-based visual place recognition task also falls into this category.

Particular instance retrieval is difficult due to the wide range of variability between the query and the reference dataset (*e.g.* changes of a viewpoint, illumination, and occlusion). Building image representations from local invariant features has been shown to be effective as they provide robustness to photometric and geometric changes (Arandjelović and Zisserman, 2014a; Jégou et al., 2012; Perronnin et al., 2010; Wu et al., 2009). However, not all local

features are useful for the given retrieval task. In particular, there are visual elements that non-incidentally co-occur in the image, which violate the assumption of feature independence and lead to over counting in bag-of-words representations. On the other hand, there are visual elements that frequently occur across images, that misleads the retrieval process. These phenomenon are commonly referred to as *visual burstiness*, which are shown to be detrimental to the retrieval performance (Jégou et al., 2009; Chum and Matas, 2010). Additionally, some features are less robust than others to photometric and geometric changes (Qin et al., 2013; Turcot and Lowe, 2009), requiring the discrimination of uninformative or obfuscating information.

Consequently, a large body of literature focuses on feature selection and weighting in image retrieval (Qin et al., 2013; Tolias and Jégou, 2014; Arandjelović and Zisserman, 2014a; Turcot and Lowe, 2009; Zhu et al., 2013; Chum and Matas, 2010; Jégou et al., 2009). Jégou et al. (2009) propose several reweighting strategies to penalize a descriptor that are matched to multiple descriptors in a database image (intra-image burstiness), and across the database images (inter-image burstiness). Chum and Matas (2010) model the dependency of co-occurring words, then use min-Hashing to detect them, and adjust their contribution to the similarity score. Arandjelović and Zisserman (2014a) use density in the descriptor space as a measure for distinctiveness. Our methods in Chapters 3 and 4 also take into account the visual elements that frequently occur across images, but they are discovered in a data-driven manner for the end-task.

In terms of selecting features in advance to matching in a data-driven way, our work in Chapter 3 is closely related to Hartmann et al. (2014), but with different focuses. Whereas Hartmann et al. (2014) tries to predict features that are likely to form a match, we predict features that contribute to correct geo-localization. As we show in our experiment, not all matches are useful for geo-localization. Our work in Chapter 4 is most relevant to the work of Shrivastava *et al.* (Shrivastava et al., 2011), which tries to find the importance of each feature by training a per-examplar SVM on a given query image with hard negative mining.

While this method can be effective, it is time consuming as a model is freshly trained at each query time. In constrast, we refine and organize the outcomes of geo-localizing training images offline, and use this knowledge for selecting features.

Recently, approaches were proposed to learn task-relevant features for image retrieval and place recognition using CNNs that are trained in an end-to-end manner (Arandjelović et al., 2016; Radenović et al., 2016; Wang et al., 2014; Lin et al., 2015). Notably, Arandjelović et al. (2016) proposed NetVLAD that integrates an end-to-end trainable VLAD layer, inspired by the vector of locally aggregated descriptors (VLAD) (Jégou et al., 2012) that is commonly used in image retrieval. Tolias et al. (2016b) proposed to use maximum values over all spatial location in the convolutional feature map as an image representation, which was employed in (Radenović et al., 2016).

These top-performing deep image descriptors treat columns of convolutional feature map (typically at the last convolutional layer) as local features. As demonstrated in the work of Luo et al. (2016) and Zhou et al. (2014a), the effective receptive field in deep CNNs takes only a fraction of the theoretical receptive field. This indicates that these local convolutional features do not actually contain much information about the image context as commonly perceived. At the same time, it is detrimental in practice to rely on high-level features (Yue-Hei Ng et al., 2015) or down-sample images (Arandjelović et al., 2016) since fine grain details are important for particular instance retrieval. Thus, in our work, we propose to explicitly use high-level context information to guide on where to focus on, while using low-level convolutional features that contain fine details for our image representation.

Our work (Chapter 4) is closely related to the work of Gordo et al. (2016) that use a region proposal network (Ren et al., 2015) to learn which regions should be pooled to form a global image descriptor. Their method uses an explicit region proposal loss that requires bounding box annotations for training, in addition to a ranking loss for retrieval. In contrast, components in our network are optimized under one triplet ranking loss for image geo-localization. Furthermore, their work outputs regions of interest as a bounding box,

whereas our proposed CRN produces a weighted mask that provides much more flexibility for focusing on relevant features.

2.1.2 Semantic Image Retrieval

Semantic image retrieval is the problem of retrieving images that have similar semantic meaning as the given query image. For instance, retrieving pictures of a library, or retrieving images with a sitting man on a bench.

Many of existing approaches use image category recognition datasets to learn image representations to capture semantic meanings in an image (Sablayrolles et al., 2017; Gong and Lazebnik, 2011; Krizhevsky and Hinton, 2011). Others adopt an image classification pipeline to parse and encode semantic information in the image (Xie et al., 2015). More recent approaches use *scene-graphs* to represent an images as a graph using the detected objects as well as their pairwise relationships (Johnson et al., 2015; Xu et al., 2017). They use a graph inferencing technique to ground the graph to the database images.

One of the major bottleneck in semantic image retrieval is the severe intra-class variation and inter-class similarity in semantic categories. It becomes increasingly hard to find a distinctive representation when the classes become visually nearly indistinguishable as the number of classes increases (Qian et al., 2015). Thus, we address this issue by apply a divide and conquer (Tu, 2005) strategy to dedicate different CNNs to separable subproblems in Chapter 5.

2.2 Visual Place Recognition

Visual place recognition is the problem of recognizing a place depicted in a query image by using a large database of geo-tagged images. It has numerous applications including augmented reality (Middelberg et al., 2014), robot navigation (Lim et al., 2012; Cummins and Newman, 2008), adding and refining geo-tags in image collections (Hays and Efros, 2015; Zamir et al., 2014), large-scale 3D reconstruction (Crandall et al., 2011), and photo editing

(Zhang et al., 2014). There are two main categories in place recognition for street-level input images: image retrieval-based methods and 3D structure-based methods.

The image retrieval-based methods approximate the geo-location of a query image by identifying the reference images depicting the same place (Arandjelović et al., 2016; Chen et al., 2011a; Hays and Efros, 2015). Our approach (Chapters 3 and 4) falls into this category of approach. These methods have the advantage of scalability, especially when used with efficient indexing schemes, such as product quantization (Jegou et al., 2011). A closely related approach is estimating the location through voting of geo-location tags associated with local features (Zamir and Shah, 2010; Vaca-Castano et al., 2012) to get a more accurate estimate than the location of the most similar database image. In this case, the nearest neighboring local features are exhaustively searched in the database for each local features in the query image, which is computationally expensive. Note that all these approaches by themselves are not capable of estimating the accurate camera pose of the query.

The 3D structure-based approaches cast the problem as a 2D-to-3D registration task (Sattler et al., 2015, 2012, 2011; Hao et al., 2012; Irschara et al., 2009; Li et al., 2012), based on a 3D model built from database images. These methods can estimate the full camera pose of the query image. However, they are limited to places with a dense distribution of reference images, and require high maintenance cost. Some train CNNs to estimate the camera pose directly from an input image (Kendall et al., 2015), however, as they encode 3D structure implicitly, it has the same problem of high maintenance cost as the model needs to be re-trained when there is an update in the database.

Combining the best of both approaches, there have been efforts to recover the 6DOF pose from the retrieved reference images. One of the earliest work (Robertson and Cipolla, 2004) uses a map associated with the database images to estimate the pose. Zhang and Kosecka (2006) estimates the camera pose from two or more related reference images. Recently, Sattler et al. (2017) proposed to build a local 3D model from the shortlist of retrieved images and use it to accurately estimate the pose of the query image.

Our work focuses on visual place recognition cast as an image retrieval task and we limit our scope to city-scale place recognition (Chapters 3 and 4). For visual place recognition, challenges come not only from photometric and geometric changes between the query and the reference images, but also from severe visual overlap with irrelevant images due to ubiquitous and transient visual elements (*e.g.* pedestrians, cars, billboards, trees). We address these challenges through an adaptive image representations using data-driven feature selection (Chapter 3) and learned contextual feature reweighting (Chapter 4). In the following, we discuss the most relevant work to ours.

2.2.1 Dealing with Geographically Ubiquitous Visual Elements

Features extracted from objects that are geographically ubiquitous can introduce obfuscating cues into the place recognition process. For example, generic windows, fences, and trees, appear in many different geographical places, and are easily matched to other instances of the same type.

There has been efforts attempting to select features that are *geographically discriminative* by taking advantage of geotags in the database. Schindler et al. (2007) build a vocabulary tree using only features that are unique to their location. Rather than finding features unique to a specific landmark, Doersch et al. (2012) find image patches that also occur frequently in a geographical region and are unique with respect to other geographic regions. Ubiquitous elements can also mislead the geometric verification process that retrieval approaches typically use as a post-processing step for refining the shortlist. Due to ubiquitous elements, the number of inlier matches between a query and a non-related image can be higher than that between the query and its related images. The problem is typically caused by generic structures with repetitive patterns. To address this problem, Sattler et al. (2016) propose a feature weighting scheme that analyzes images in the shortlist using their associated GPS-tags to detect and down-weight such geometric bursts.

While our models (Chapters 3 and 4) are not explicitly trained to discriminate geographically ubiquitous visual elements, it automatically discovers them and adjust their weights in our image representation.

2.2.2 Dealing with Transient Elements

Some visual elements change their appearance over time. For instance, trees change their leaves according to the season, pedestrians and vehicles come and go, billboards change and are often animated. Such transient visual elements make place recognition very challenging.

To address this problem, some approaches synthesize different views from database images to minimize the geometric variability in order to limit the overall appearance variability (Torii et al., 2015a; Sibbing et al., 2013; Aubry et al., 2014). The downside of these approach is scalability, because the number of reference images has increased. However, this problem could be alleviated via using efficient nearest neighbor search schemes (Jegou et al., 2011). On the other hand, others aim at learning local features that are *stable* over time (Chen et al., 2017b; Linegar et al., 2016; Naseer et al., 2014). However, these method only address half of the problem. Stable features may include the features from ubiquitous objects discussed earlier which distract the recognition system, such as generic man-made objects.

We demonstrate that our data-driven approach considers both ubiquitous and transient visual elements (Chapters 3 and 4).

2.2.3 Use of Context Information

Contextual information has been widely used for object recognition (Oliva and Torralba, 2007; Rabinovich et al., 2007; Yu and Koltun, 2016; Bell et al., 2016; Girshick et al., 2016; Kantorov et al., 2016). However, the use of context is significantly limited for image retrieval-based place recognition, such as building versus non-building. Mousavian and Košecka (2015) and Naseer et al. (2017) used semantic segmentation to filter out local features. Sünderhauf et al. (2015) use pre-trained object proposals to detect landmarks. All these approaches

are based on the assumption that reliable features are likely to occur on man-made objects such as buildings. However, features from man-made objects often contain ubiquitous visual elements (Sec. 2.2.1) and do not necessarily contribute to the correct localization. For example, generic windows and fences, as well as repetitive structure on the building facades misleads the localization process, similar to the *visual burstiness* phenomenon discussed in Sec 2.1.1. Torii et al. (2015b) adjust the weights for features occurring in repetitive structure to address this issue. Arandjelović and Zisserman (2014b) embedded semantic information to disambiguate matching between local features, but it is not used for discriminating useful image content.

In contrast, our work (Chapter 4) uses a learned top-down context information to guide our representation to focus on useful image content in a strictly data-driven manner.

2.2.4 Data-Driven Notion of Useful Visual Elements for Place Recognition

Our work (Chapters 3 and 4) is motivated by the work of Knopp et al. (2010), which refines the database by removing features that match to faraway places. The drawback of this approach is that the computational cost for the refinement is quadratic to the number of database, which is not suitable for a very large database. On the other hand, some methods train a classifier per each geographic location such that features are naturally weighed differently for each specific location (Gronat et al., 2013; Cao and Snavely, 2013; Weyand et al., 2016; Chen et al., 2017b). However, these approaches require a model to be trained for all possible locations in the dataset.

Our approach also falls into the category of data-driven methods. However, in contrast to other methods, we either need only a single inference on database images (Chapters 4), or just on the query image (Chapters 3).

2.3 Automatic Training Data Generation

There have also been efforts to automatically generate training data for image retrieval and place recognition for CNNs. Radenović et al. (2016) exploit 3D reconstruction for selecting training data and Gordo et al. (2016) use landmark graphs obtained from pairwise matching of images in the dataset. However, both their methods require a dense image distribution in order to construct 3D models or scene graphs. We acquire training data similarly to the work of Arandjelović et al. (Arandjelović et al., 2016), using GPS-tags (Chapters 3 and 4). As geo-location itself is not sufficient to determine image overlap due to different camera orientations and occlusions, they choose positive images based on the current representation during learning. This depends on the quality of the learned representation, and does not contribute much to the current network status. On the other hand, we not only use GPS-tags, but use geometric verification to once verify positive images and refine them, thus reducing the memory and compute requirements (Chapters 3 and 4). Also, while these methods (Radenović et al., 2016; Arandjelović et al., 2016) perform periodic full retrieval for hard negative mining, we introduce within-batch hard negative mining for image geo-localization which is computationally less expensive (Chapters 4). This is similar to within-batch hard negative mining in (Wang and Gupta, 2015), and stochastic sampling used in (Wang et al., 2014). However, we use GPS-tags to determine negatives.

2.4 Tree Structured Deep Neural Networks

Our method presented in Chapter 5 takes the hierarchical mixture of experts approach (Bishop and Svenskn, 2002; Jordan and Jacobs, 1994), where each expert in the tree structure learns to handle splits of the input space. In light of recent advances in deep neural networks, many researchers have revisited the mixture of experts for various computer vision applications, such as visual category recognition (Yan et al., 2015; Ahmed et al., 2016), lifelong learning (Aljundi et al., 2017), face alignment (Tuzel et al., 2016), and segmentation (Hiramatsu et al.,

2018). In particular, our method adopts the generalist and specialist model from the work of Hinton et al. (2015), which is similar to the mixture of experts in the sense that each specialist focuses on a confusable subset of the classes, but it has a generalist that can handle classes that are not handled by the specialists. It also does not require the training of a gating function, allowing models to be trained in parallel. Such approach has a distinct advantage compared to the widely-used ensemble averaging paradigm (Sollich and Krogh, 1996; Zhou et al., 2002; He et al., 2016; Ju et al., 2017) in terms of dealing with large intra- and inter-class variation, as each specialist model is trained on data which consists of examples from a highly confusable subset of the problem. While one way of defining the area of specialties is using a semantic hierarchy (Goo et al., 2016; Deng et al., 2011), we focus on unsupervised approaches. Similar ideas to Hinton et al. (2015) were proposed in (Yan et al., 2015; Murthy et al., 2016; Ahmed et al., 2016; Warde-Farley et al., 2014). Yan et al. (2015) allow the coarse categories to overlap. Murthy et al. (2016) extended this concept to a tree structure with more than two levels of hierarchy. Ahmed et al. (2016) take an iterative approach to jointly optimize the grouping of class categories and the model parameters rather than having a fixed partition before training the specialists. In the context of transfer learning, Srivastava and Salakhutdinov (2013) proposed a method for learning to organize the classes into a tree hierarchy for dealing with limited data.

However, all these approaches partition the input space by grouping categories, while our method partitions the feature space that captures high-level appearance information regardless of class membership, based on the observation that there are visually drastically different sub-classes within each class. This also frees our method from the risk of misclassification in specialties (due to large intra-class variation) from which classifiers cannot recover for class-based grouping schemes (Yan et al., 2015; Murthy et al., 2016; Ahmed et al., 2016; Warde-Farley et al., 2014). Moreover, our method only invokes a limited number of models during testing, which leads to significant gains in computational efficiency.

In contrast to organizing multiple CNN models, there have been efforts to separate visual features in a single CNN in a tree structure (Kim et al., 2017b; Ahmed and Torresani, 2017; Murdock et al., 2016; Sabour et al., 2017; Li et al., 2017). This is especially useful for parallel and distributed learning as demonstrated in Kim et al. (2017b), where disjoint sets of features, as well as disjoint sets of classes are automatically discovered. In the same spirit of parallelization, but on a much larger scale, Gross et al. (2017) deal with a mixture of experts model that does not fit in the memory. Similar to their work, our learned submodels are local in the feature space, and the image-to-model assignment is determined by the distance of the image to the corresponding submodel cluster center.

2.5 Scene Category Recognition

Numerous work has been done on scene categorization as being one of the fundamental problems in computer vision (Oliva and Torralba, 2001; Lazebnik et al., 2006; Quattoni and Torralba, 2009; Xiao et al., 2010; Zhou et al., 2018; Guo et al., 2017; Wang et al., 2017a; Xiao et al., 2014; Khan et al., 2017). Our work is related to recent approaches that leverage object information within the scene (Cheng et al., 2018; Wang et al., 2017b; Dixit et al., 2015; Dixit and Vasconcelos, 2016; Herranz et al., 2016; Zhou et al., 2014b). However, we do not explicitly detect objects using pre-trained networks, requiring additional labels, or perform rigorous clustering offline to find such visual elements (Juneja et al., 2013; Wu et al., 2015). Instead, we let the network capture such information during the end-to-end training process through a network architecture that accounts for objects that can freely move within the scene. Global order-less pooling of convolutional features has a high degree of invariance for encoding local visual elements such as objects. In this way, high level convolutional filters perform like an object detector as analyzed in (Zhou et al., 2014a; Bau et al., 2017). Furthermore, we also leverage global ordered pooling representations which preserve coarse spatial information (Mousavian and Kosecka, 2015).

CHAPTER 3: PREDICTING GOOD FEATURES FOR IMAGE GEO-LOCALIZATION

Image geo-localization is the process of determining the position from which an image is taken w.r.t. a geographic reference (Zamir and Shah, 2014). The recent availability of large scale geo-tagged image collections enables the use of image retrieval frameworks to transfer geo-tag data from a reference dataset into an input query image. Applications of these capabilities include adding and refining geotags in image collections (Hays and Efros, 2008; Zamir et al., 2014), navigation (Lim et al., 2012), photo editing (Zhang et al., 2014), and 3D reconstruction (Frahm et al., 2010). However, geo-localization of an image is a challenging task because the query image and the reference images in the database vary significantly due to changes in scale, illumination, viewpoint, and occlusion.

Image retrieval techniques based on local invariant image features (Lowe, 1999) provides robustness to photometric and geometric changes (Li et al., 2010; Zamir and Shah, 2010). However, not all local features are useful for geo-localization (Knopp et al., 2010). For example, features extracted from transient scene elements (pedestrians, cars, billboards) and ubiquitous objects (trees, fences, signage) can introduce obfuscating cues into the geo-localization process. Many approaches have been proposed to address this issue by focusing on the *uniqueness* of a feature by removing and reweighting non-unique features within the reference data (Knopp et al., 2010; Schindler et al., 2007) or in the query image (Arandjelović and Zisserman, 2014a). Indeed, unique features are helpful, but a non-unique feature may actually help increase the chance of correct localization, either by themselves or in combination with others.

We exploit a data-driven notion of good features for geo-localization (Kim et al., 2015). That is, we aim to foster features having relatively high matching scores in correct localization outcomes, in contrast to their relatively low score for negative outcomes. Further, we cast

feature score prediction as a classification problem, assuming the characteristics are shared in a reasonably-scaled geographic region. We use a separate set of geo-tagged Internet images to generate training data, computing matches against database images. To cope with noise and high intra-class variation among the training data, we adopt bottom-up clustering techniques for visual element discovery (Doersch et al., 2013, 2012) that involve iterative training of linear support vector machines (SVM). At the query phase, the algorithm selects features in a query image prior to the geo-localization process by accumulating predictions from the bank of linear SVMs. Our results show that using only features that are predicted as useful improves the localization accuracy while reducing the computational cost significantly.

The feature representation for such a task should not only be robust to photometric and geometric changes, but also have a high discriminative power as we want to learn features over a large area, *e.g.* a city. Therefore, we avoid using low-level features for learning, which are hard to be discriminative over a large area. We propose the per-bundle vector of locally aggregated descriptors (PBVLAD) for feature representation, where each maximally stable (MSER) (Matas et al., 2004) region is described with a vector of locally aggregated descriptors (VLAD) computed from multiple scale-invariant features detected within the region. This allows us to represent multiple features with a fixed-size vector such that it can be used in various classification methods such as SVM. We show in the experiments that this feature representation leads to significant improvement over low level features in both learning to predict features and retrieving images.

The main contribution of this chapter is two-fold: (1) We offer a way to predict features that are good in a data-driven sense for geo-localization in a reasonably-scaled geographic region (*e.g.* a city). We show that by selecting features based on predictions from learned classifiers, geo-localization performance can be improved. (2) We propose the per-bundle vector of locally aggregated descriptors (PBVLAD) as a novel representation for a bundled feature that is effective for both learning to predict features and image retrieval.

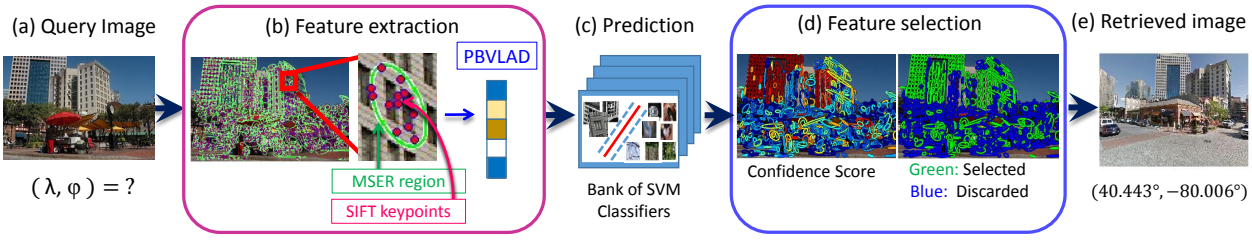


Figure 3.1: Overview of our approach. Given an input query image with unknown geo-location (a), MSER regions and SIFT keypoint form bundled features (Wu et al., 2009), which are then represented by PBVLADs (b). Features go through a pre-trained bank of SVMs that outputs binary predictions about a feature being "good" for geo-localization (c). Predictions are accumulated to compute confidence scores for each feature (d, left). Features with high scores are selected for geo-localization (d, right). A retrieved geo-tagged image is shown in (e).

3.1 Proposed approach

The overview of our approach is shown in Figure 3.1. In this section, we first introduce our proposed feature representation for image retrieval and training classifiers (Sec. 3.1.1). We then illustrate our training framework for automatically generating training data and training a bank of SVMs for predicting good features for geo-localization (Sec. 3.1.2).

3.1.1 Per-bundle VLAD for feature representation

We want to identify parts of an image that are useful for geo-localization, using a discriminative classification method such as SVM. However, it is a hard problem to learn such characteristics given a low level description of a corner or a blob. Thus, we propose per-bundle vector of locally aggregated descriptors, namely PBVLAD. The key idea is to use groups of low level features, and describe them in a vector with a fixed-size that allows it to be compared in standard distance measures and enables it to be used for various classification methods.

The concept of a *bundled feature* was proposed by Wu *et al.* (Wu et al., 2009) for retrieving partial-duplicate images. By bundling multiple SIFT features detected in the same MSER region, the discriminative power is increased while still being repeatable, as both components are robust to photometric and geometric changes. The original representation

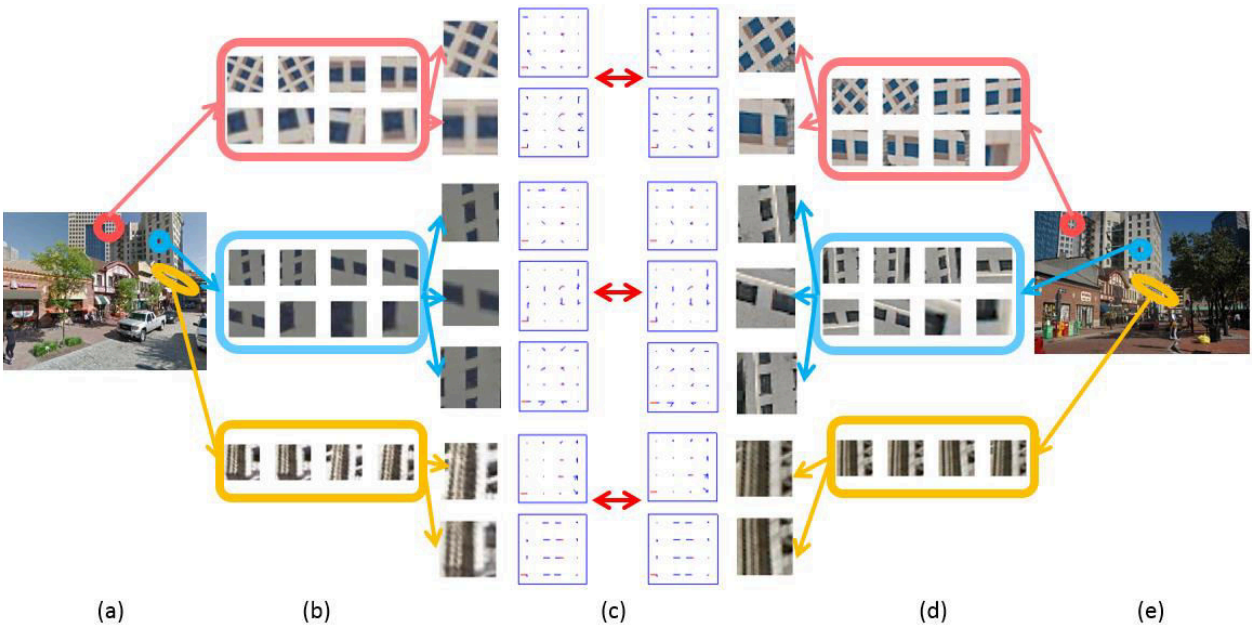


Figure 3.2: PBVLAD representations of corresponding bundled features. (a,e) Two different images depicting the same place. (b,d) Multiple SIFT features are bundled within MSER regions. (c) Each bundle is represented with VLAD. We follow the visualization scheme of (Jégou et al., 2012) where subvectors are represented in a 4x4 spatial grid with red representing negative values. Note that only non-sparse blocks that correspond to overlapping visual words of two bundles are visualized due to space limits.

was a concatenation of quantized SIFT features, which changes in length as a MSER region can contain different number of SIFT features. The similarity between two bundled features was measured by computing the intersection between them. In this paper, we propose to describe a bundled feature with a vector of locally aggregated descriptors (VLAD) (Jégou et al., 2012). This representation produces sparse vectors with a fixed-size that are convenient for comparing distances and training classifiers such as SVM. Compared to the bag-of-words (BoW) representation, VLAD can have a much smaller dimension while maintaining high discriminative power, and it can be further quantized without significant loss in performance. Note that Min-hash sketches can also provide a compact representation (Chum et al., 2009), but they have a comparably low recall and a limited number of applicable classification methods as standard distance measures cannot be applied.

Let R and S denote the MSER regions and SIFT features detected in image I, respectively. Each MSER region $r \in R$ contains a set of SIFT features $B \subset S$ that are detected within that region $B = \{s = (d, l) | l \in r\}$, where d and l denote the descriptor and the location of the SIFT feature. B is called a bundled feature (Wu et al., 2009). For a bundled feature B_a, its associated SIFT features $s_a = (d_a, l_a) \in B_a$ are each assigned to a visual word of a coarse vocabulary W via nearest neighbor search such that $NN(d_a) = \underset{w}{\operatorname{argmin}} ||d_a - c^w||$, where c^w is the centroid of the visual word w. The subvector of per-bundle VLAD that corresponds to the visual word w, denoted as p_a^w, is obtained as an accumulation of differences between d_a's that are assigned to w and the centroid c^w:

$$p_a^w = \sum_{d_i : NN(d_i) = w, d_i \in B_a} \frac{d_i - c^w}{\|d_i - c^w\|}. \tag{3.1}$$

As proposed in (Delhumeau et al., 2013), we normalize the differences (*i.e., residuals*), so that each contribution of SIFT descriptor d_i to the vector p_a^w are equal. This is to limit the effect of possible noise, although bundled features are robust to photometric and geometric changes. The final representation is the concatenation of the vectors p_a^w, that is, $p_a = \left[p_a^1, p_a^2, ..., p_a^{|W|}\right]$, followed by L_2 normalization. We tested multiple normalization schemes (Arandjelović and Zisserman, 2013; Jégou et al., 2012), but the combination of residual- and L_2- normalization performed the best on our data. The PBVLAD representation of corresponding bundled features are visualized in Figure 3.2. Henceforth, the term *feature* will refer to PBVLAD representation of a bundled feature.

Similarity metrics. The similarity between two PBVLAD is computed as their dot product $M(p_a, p_b) = p_a \cdot {p_b}^T$. Figure 3.3 depicts the matched feature regions of two corresponding images. We define the *matching score* f of a PBVLAD feature p_q in a query image I_q to a reference image I_r as the maximum possible similarity between p_q and features in I_r:

$$f(p_q, I_r) = \max_{p_r \in I_r} M(p_q, p_r). \tag{3.2}$$

Figure 3.3: Matching with PBVLAD with similarity threshold 0.5.

The *image similarity* Sim between a query image I_q, and the reference image I_r becomes the sum of matching scores of individual features $p_q \in I_q$ with respect to I_r:

$$Sim(I_q, I_r) = \sum_{p_q \in I_q} f(p_q, I_r). \tag{3.3}$$

We use the image similarity measure defined above to retrieve reference images that best matches the query image.

For efficient nearest neighbor search in the reference data, we reduce the dimension of the raw PBVLAD using principal component analysis (PCA). Instead of performing PCA on a whole vector, we do it on a per-visual-word basis by performing PCA on subvectors p^w that are generated from each visual word w. We do this in order to preserve the characteristics of each visual words that might be lost due to the overall sparsity of the vector. In our implementation, a coarse vocabulary of 128 visual words was used, yielding 16,384-dimensional raw PBVLAD. The dimension is then reduced to 2,048 by performing PCA on 128 visual words and taking the top 16 components of each. Note that PBVLAD matching can be efficiently indexed using product quantization (Jegou et al., 2011).

Figure 3.4: Initial training data generation. Positive and negative training examples are depicted in green and blue, respectively.

3.1.2 Predicting good features for geo-localization

In this section, we first outline our method for automatically generating the training examples. We then describe our classification model and the details of training procedure.

3.1.2.1 Automatic training data generation

Given a set of geo-tagged images $\mathcal{I}_t = \{I_t\}$, we want to *automatically* generate good/bad training examples of features for geo-localization using only their associated GPS locations. Rather than making assumptions about good and bad features for geo-localization, we want to find them in a data-driven way. This enables our method to adapt to various geographical regions. For each image in the training set, we retrieved top $n = 100$ images from the reference set $\mathcal{I}_r = \{I_r\}$ using the image similarity defined in Eq. 3.3. We investigate whether a feature in a training image $p_t \in I_t$ is *explicitly* contributing to the correct retrieval of the ground

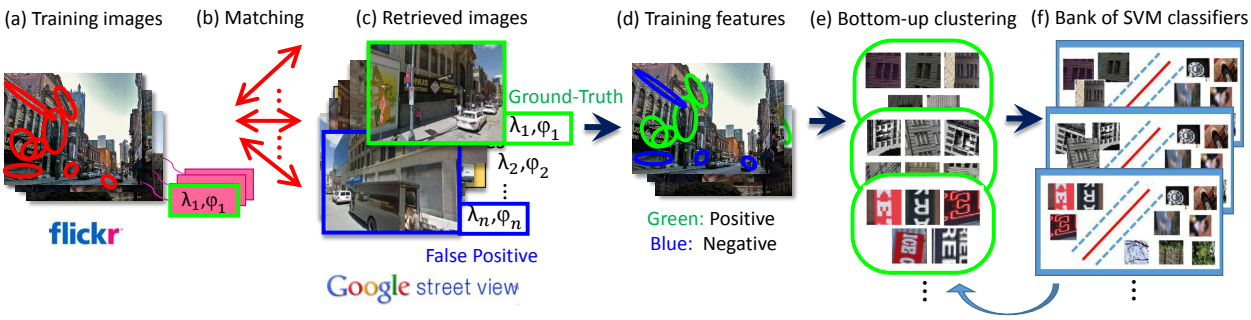


Figure 3.5: Overview of our training framework. For all training images that have GPS-tags (a), we retrieve top n images from the reference set (b-c). Positive labels are assigned to features that have higher matching score in the ground-truth reference image than in the falsely retrieved reference images, with a margin greater than *thres*. Negative labels are assigned in a similar manner (d). To handle noise and high intra-class variation, we use a bottom-up clustering technique, refining the positive set as well as training SVMs iteratively (e-f).

truth images I_{GT}. To this end, we compare a feature's matching score to a ground truth reference image $f(p_t, I_{GT})$ against the matching score to a falsely retrieved image $f(p_t, I_{FP})$. Given that the overall image similarity between two images is the sum of individual matching scores (Eq. 3.3), this comparison helps us differentiate good features based on their individual contribution. We define the ground images I_{GT} as reference images that are within 50 meters from the given GPS location and passed geometric verification (Philbin et al., 2007) w.r.t. the training image by fitting a fundamental matrix (Hartley and Zisserman, 2003). For falsely retrieved images I_{FP}, we took reference images that are retrieved within the top n (n =100), and at least 270m away from the given GPS location. This accounts for both user-provided geo-tag errors and the fact that large, symmetric buildings are often observable from extended areas. If the difference between the two values $|f(p_t, I_{GT}) - f(p_t, I_{FP})|$ is greater than a certain threshold, we include the feature into the training set, assigning a positive label when $f(p_t, I_{GT}) > f(p_t, I_{FP})$, and a negative label otherwise. If there are multiple I_{GT} and I_{FP}, we take the maximum of each $f(p_t, I_{GT})$ and $f(p_t, I_{FP})$ for comparison. This process is depicted in Figure 3.5 (a-d) and provides the initial positive and negative training feature set for data-driven visual component discovery.

3.1.2.2 Closed-loop training of SVM classifiers

The automatic labeling approach above can sometimes generate contradictory labels for features with similar appearance. This commonly occurs in visual elements that appear on both transient and static objects. In Figure 3.4, for example, text on buses (b) and t-shirts (e) is assigned a negative label, while text on buildings and store signs (d) belongs to the positive set. A limited field-of-view overlap between a training image and a ground truth image can also lead to such contradictory labeling. Windows on the same building, for instance, can be assigned to different labels due to their visibility in the ground-truth reference image I_{GT}. Such contradictory labeling on similar features limits the prediction accuracy.

On the other hand, there exists high intra-class variation in both the positive and negative classes: Windows have different appearances from text, for example, yet features from both appear in the same class. Training a single classifier over the entire data may be negatively affected by such intra-class variation.

To solve the problems of contradictory labeling and intra-class variation, we perform bottom-up clustering (Doersch et al., 2012) on the initial training feature set. By doing so, we obtain clusters of training examples whose appearances and labels are most consistent, as well as a bank of linear SVM classifiers that are trained within each cluster. Each training example constructs a cluster by finding k nearest neighbors in the training set. Redundant sets whose top ranked elements overlap with existing sets are eliminated. If a cluster has a high ratio of negative labels, the negative examples in that cluster are assigned to the final negative set $\mathcal{N}$ and the positive ones are discarded.

For the remaining clusters C_i, a linear SVM is iteratively trained on the positive examples in each cluster, using $\mathcal{N}$ as the negative set for hard negative mining (Figure 3.5(e-f)). As the SVM uses its true-positive firings for the re-training in the iterative procedure, clusters are left with features having consistent appearances and labels. Similar to (Doersch et al., 2012), the clusters and $\mathcal{N}$ are divided into three sets to avoid overfitting. We only keep the SVM classifiers with an accuracy rate greater than 0.8. Finally, we remove redundant

Figure 3.6: Top elements in the final clusters with a high ratio of positive labels. Each half row corresponds to different clusters.

classifiers whose weight vectors have a high cosine similarity with that of other classifiers as in (Juneja et al., 2013). Examples of top elements in $\mathcal{C}_i$ are shown in Figure 3.6. Figure 3.7 shows elements in $\mathcal{N}$, which are aligned according to their initial clusters. Interestingly, although our approach makes no assumption on features that are useful for geo-localization, we can observe semantic relationships emerge through the learning process. Namely, windows, characteristic wall patterns, and letters on signage are detected as positive elements, while features from trees, people, car wheels, pavements, and edges are considered as negative elements.

In the querying phase, we feed query image features into the bank of linear SVM classifiers. We accumulate predictions from each classifier to compute the confidence score of a feature being good for geo-localization (Figure 3.10 (b)), weighting them using the *discriminativeness* (Singh et al., 2012) of the classifier, which is the ratio of number of firings in its cluster $\mathcal{C}_i$ over that in the entire training set, in order to compensate for the distribution of visual elements that each cluster spans. We discard features with a low confidence score and keep only the remaining features for performing geo-localization (Figure 3.10 (c)).

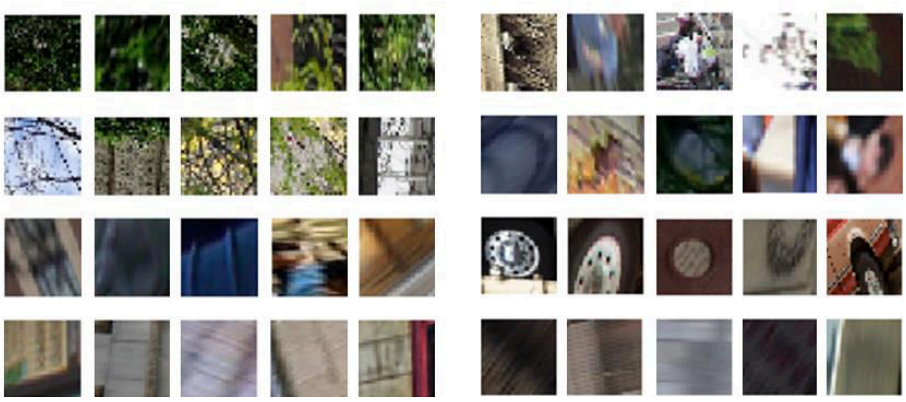


Figure 3.7: Final negative set elements aligned according to their initial clusters. Each half row corresponds to different clusters.

Implementation details. Before comparing $f(p_t, I_{GT})$ and $f(p_t, I_{FP})$, we normalize the matching scores by multiplying $\frac{1}{\max(f)}$ to compensate for a non-uniform distribution of features. For training and prediction, we separated features into three scale levels based on the size of the MSER, as we observed that the distribution of positive and negative PBVLAD features varies in different scales. The number of SVM classifiers used in each level were 35, 150, and 25, in an ascending order of the MSER size.

3.2 Experiments

We first evaluate our proposed pipeline with feature selection based on PBVLAD for image geo-localization. We quantitatively compare it with various baselines to demonstrate its efficacy. We also show qualitative results and failure cases to illustrate the performance of our method. Then, to further assess the benefit of our proposed PBVLAD, we evaluated it on a standard image retrieval benchmark.

3.2.1 Image Geo-localization

Dataset. For the reference image set $\mathcal{I}_r$, we used 27,520 geo-registered Google Street View images covering the Pittsburgh (U.S.) area, which is generated from the same panorama

images that is used for the Pittsburgh portion of the reference set in the work of Zamir and Shah (2010). These images contain 8 overlapping perspective views extracted from the spherical panoramas in two different pitch directions, to capture both eye-level street views and the higher parts of the building in urban environments. This setting is similar to those used in (Doersch et al., 2012; Gronat et al., 2013; Torii et al., 2015b). The test image set $\mathcal{I}_q$ was formed by 145 Internet collection images from the query set of Zamir and Shah (2010) with manually verified GPS-tags that are taken in Pittsburgh. The co-located GPS-tagged training image set $\mathcal{I}_t$, comprising positive and negative training data $\mathcal{C}_i$'s and $\mathcal{N}$ for learning, was downloaded from Flickr and consisted of 850 images that were successfully registered to the geographically nearby images in $\mathcal{I}_r$ through geometric verification.

Results. We compare the proportion of correctly localized images among a ranked list of the top n candidates. All of our results are without post-processing such as geometric re-ranking (Philbin et al., 2007). We consider an image to be localized if it is within 35m from the ground truth location. For a baseline, we compare with our implemented version of (Zamir and Shah, 2010). We also compare a variant of (Zamir and Shah, 2010) with SIFT feature selection by pre-trained linear SVM in a procedure similar to our selection of PBVLAD features (SIFT Select).

Figure 3.8 depicts how our systems with selected PBVLAD (PBVLAD Select) and all PBVLAD (PBVLAD All) consistently outperform the baseline methods. Feature selection is more successful in PBVLAD than SIFT. The performance of using selected features is consistently better than using all features in PBVLAD, whereas this behavior alternates when considering SIFT features.

The recall at the top retrieved result ($n = 1$) is displayed in Table 3.1. Our method achieves a recall of 64.83% using all features and improves to 68.28% with selected features, while the best baseline method (SIFT Select) obtains 49.66%. We also tested the performance of the system using the same number of PBVLAD features as our selection framework, but that are picked randomly (PBVLAD Random). Its poor recall rate supports the effectiveness

Table 3.1: Proportion of correctly localized images at top 1

Method	% Correct
PBVLAD All	64.83
PBVLAD Select	**68.28**
PBVLAD Random	33.38
PBVLAD Select$^{\complement}$	19.31
SIFT All (Zamir and Shah, 2010)	49.66
SIFT Select	46.90
Chance	0.20

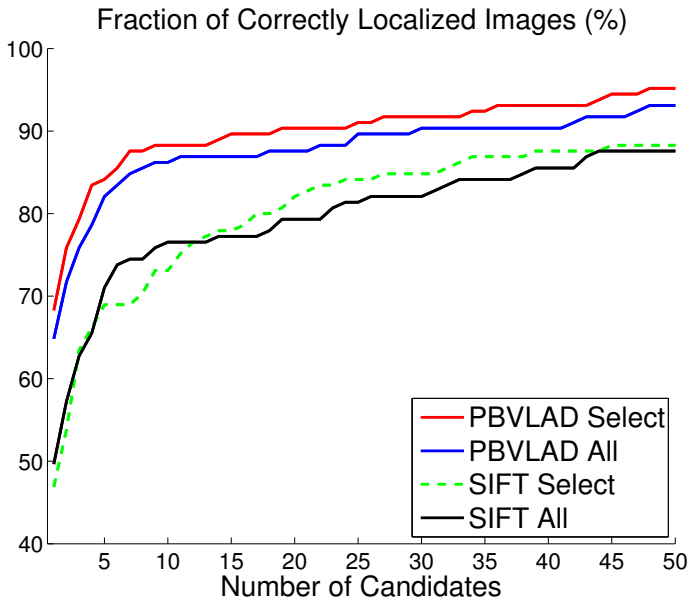


Figure 3.8: Geo-localization performance

of our selection mechanism, illustrating how simply selecting fewer features does not generally improve the performance. Moreover, we also tested with the features that are *not* selected by our framework (PBVLAD Select$^{\complement}$) to illustrate how discarded features are in general detrimental to the geo-localization. The random chance of retrieving correct images is 0.2 %, which reflects difficulty posed by the dataset.

Figure 3.9 shows examples of our results using PBVLAD Select. The top four retrieved images are shown for each query image. As can be seen, our method retrieves correct reference

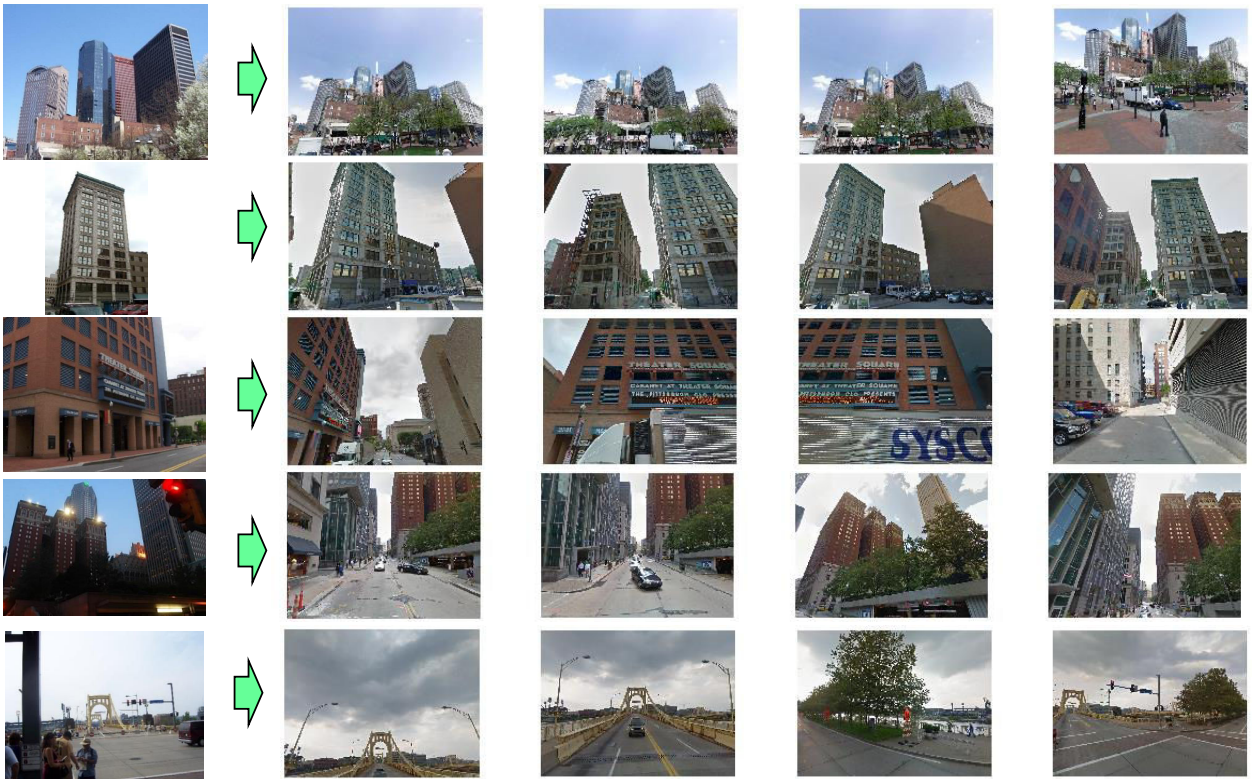


Figure 3.9: Example result (left) Query images, (right) Top four retrieved images using our proposed PBVLAD with feature selection. Query images are of various sizes.

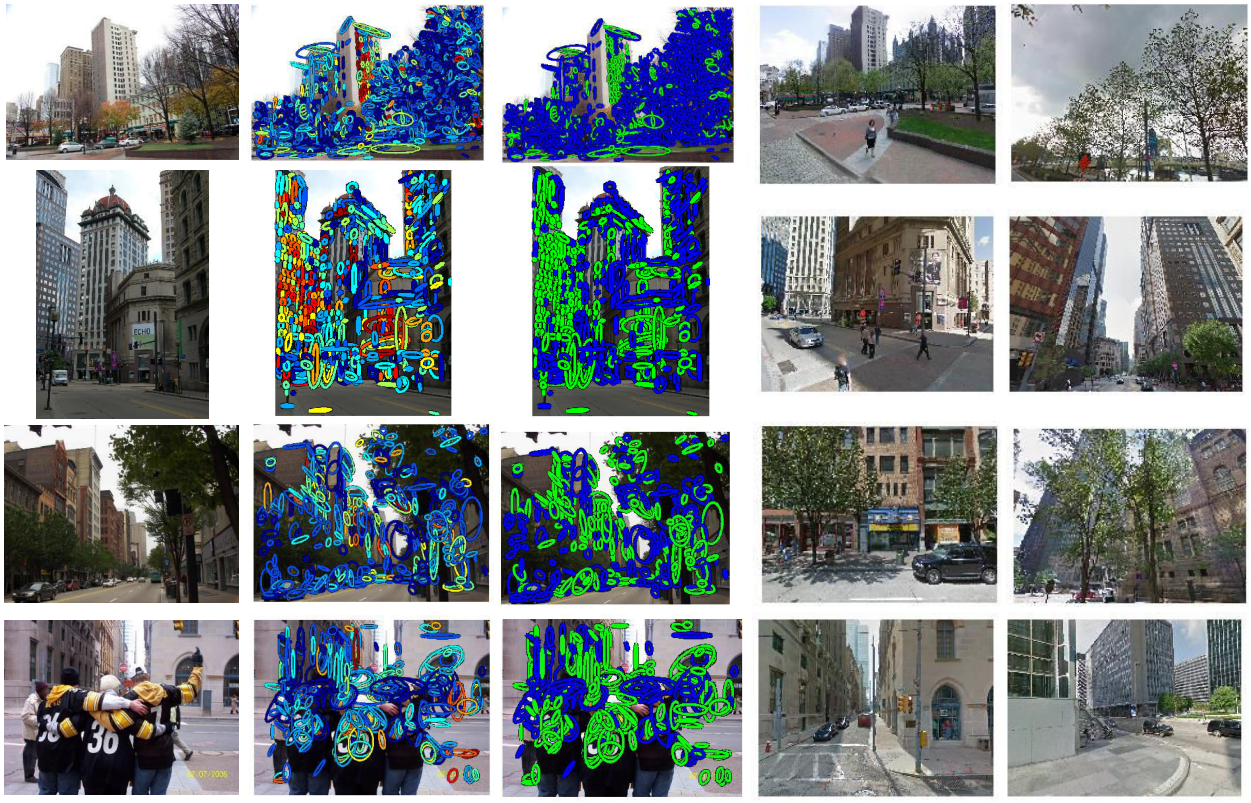


Figure 3.10: Qualitative comparison of retrieved image using selected PBVLAD and using all of the features. (a) Query image. (b) Heat map representation of the confidence of being a good feature. (c) Selected features (green:selected, blue:discarded). (c) The top retrieved image using selected features. (d) The top retrieved image using all features.

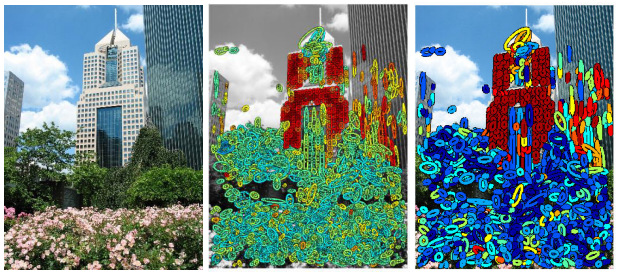


Figure 3.11: (a) Query image. (b) Heat map of maximum matching scores $\max_{I_r}(f(p_q, I_r))$ of all features p_q. (c) Confidence scores.

images despite partial occlusions and changes in viewpoint, illumination, and scale. Figure 3.10 depicts other examples where PBVLAD Select outperforms PBVLAD All.

We attribute the enhanced performance of PBVLAD-based retrieval to the increased discrimination power provided by aggregated features. Figure 3.11 (b) illustrates the maximum obtained feature similarity score for the features within a query image (a) w.r.t. the entire reference dataset. We can observe that PBVLAD features in foliage image regions are not highly matched to the reference set. Where individual SIFT features may have many similar features in the dataset, the analysis of their local ensembles is more discriminative. Moreover, our final predicted feature scores (c) illustrate how our framework discriminates good features prior to matching.

3.2.1.1 Failure cases

There are many cases where the ranked list contained the same building in the query image, but at distant locations from the ground-truth. The first and second row of Figure 3.12 show such examples. This occurs often for images depicting a large and symmetric building. In many cases, the building itself looked more similar to the retrieved image than the ground-truth reference image. Another observation is that when it comes to severe scale changes, the number of SIFT keypoints detected within the MSER region is reduced due to a lack of details. In such cases, it becomes hard to match a PBVLAD as many of its

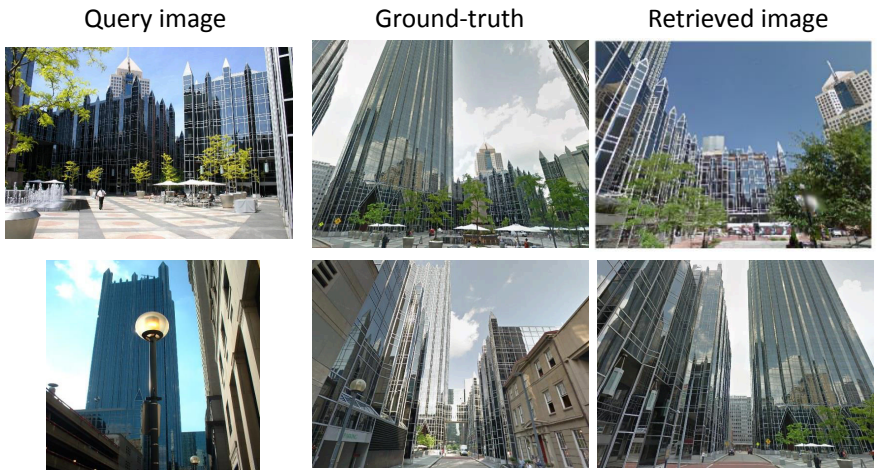


Figure 3.12: Failure cases. Retrieved images are more than 100m away from the ground-truth locations.

group members are missing. This could be alleviated by using spectral SIFT (Koutaki and Uchimura, 2014), or by only including keypoints detected within some scale range from the MSER region similar to (Chum et al., 2009).

3.2.2 PBVLAD for general image retrieval

To further assess the benefit of the PBVLAD as a descriptor for general image retrieval, we evaluated it on the Oxford5k Buildings dataset (Philbin et al., 2007) without the feature selection procedure. Table 3.2 compares our method against state-of-the-art image retrieval approaches (Eggert et al., 2014; Jégou et al., 2012), which includes VLAD, Fisher vector (FV), and a bag-of-words baseline. The evaluation was performed without dimensionality reduction for all methods. The PBVLAD shows competitive performance to other state-of-the-art descriptors. Table 3.3 shows the effect of dimension reduction using PCA. The decrease in the performance is not significant until the dimension is reduced below 12.5%. Examples of the top 20 retrieved images using the raw PBVLAD are shown in Figure 3.13. The average precisions (AP) are shown below each query image.

Table 3.2: Comparative image retrieval performance of PBVLAD on the Oxford 5k dataset. The accuracy is measured by the mean Average Precision (mAP). All descriptors are uncompressed.

Descriptor	# Vocabulary	mAP
BoW (Jégou et al., 2012)	200,000	0.364
BoW (Jégou et al., 2012)	20,000	0.319
Fisher (Jégou et al., 2012)	64	0.317
VLAD (Eggert et al., 2014)	128	0.339
PBVLAD	128	**0.369**

Table 3.3: Retrieval performance of PBVLAD on Oxford 5k dataset, before and after the dimensionality reduction using PCA. The accuracy is measured by the mean Average Precision (mAP).

	Full	Dim Reduced			
Dim	16384	8192	4096	2048	1024
mAP	**0.369**	0.364	0.334	0.264	0.210

3.3 Conclusion

In this chapter, we proposed the per-bundle vector of locally aggregated descriptors (PBVLAD) for maximally stable regions in an image. PBVLAD provides a convenient and effective representation for classification of grouped local features. Using this descriptor and a geo-tagged internet image collection, good/bad features for geo-localization were exploited with the notion of good/bad being explicitly defined in terms of the feature's contribution to the retrieval process. To remove noisy labels and be robust to the large intra-class variation, bottom-up clustering was performed, generating a bank of SVM classifiers. At the query phase, outputs of each classifiers were accumulated to select good features. The experimental results shows that not only the proposed PBVLAD by itself has a significant advantage over SIFT descriptors for image geo-localization, but also that the accuracy is further improved when using only good features predicted by our algorithm.

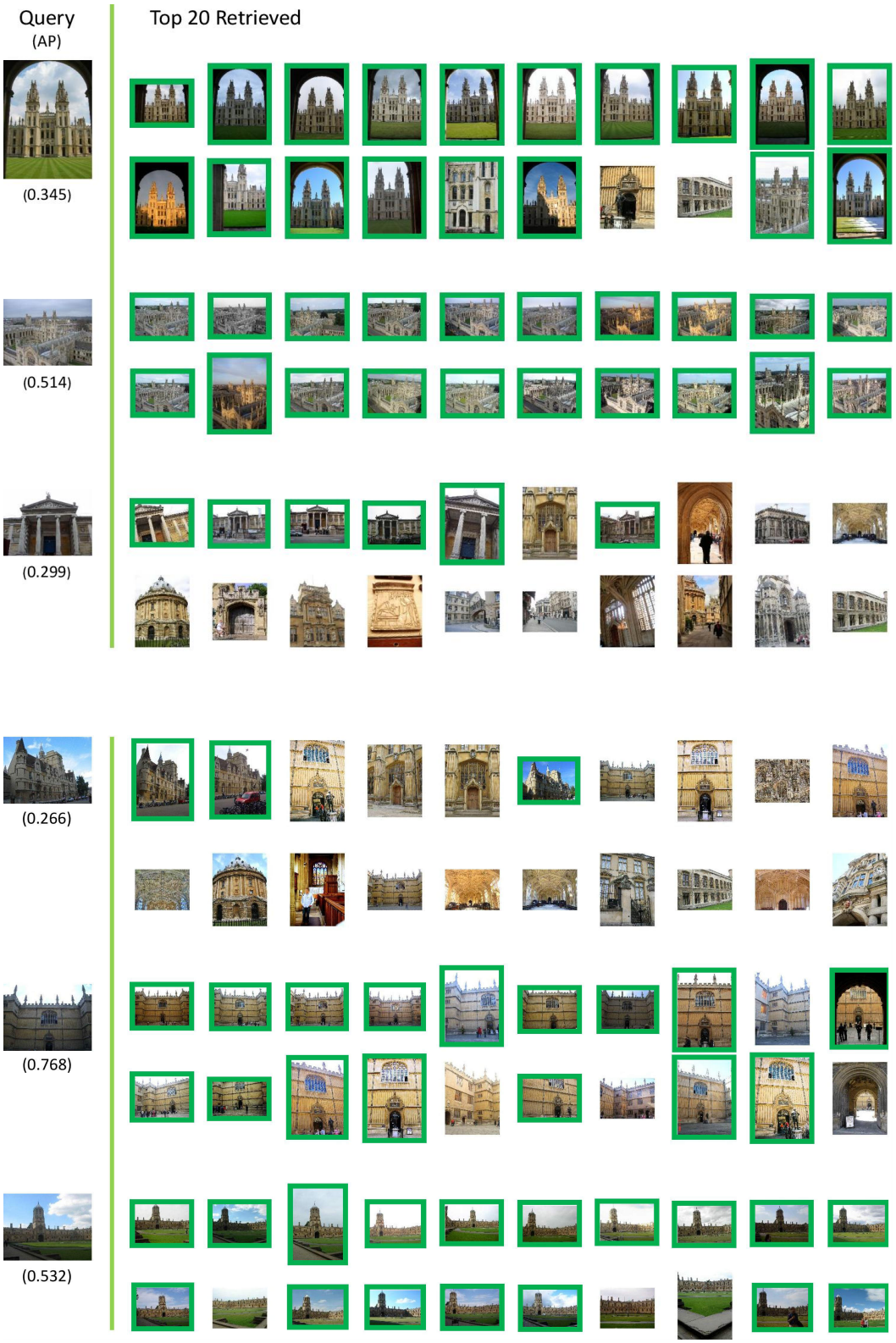


Figure 3.13: Image retrieval result on Oxford Buildings 5k dataset. (left) Query images and average precisions (AP) by our system. (right) Top twenty retrieved images using PBVLAD as an image descriptor, where the image with the highest similarity score is shown on the top left. The green boxes around retrieved images denote the correct retrieval results.

CHAPTER 4: LEARNED CONTEXTUAL FEATURE REWEIGHTING FOR IMAGE GEO-LOCALIZATION

Visual image geo-localization has been an active research area for the past decade (Arandjelović et al., 2016; Kim et al., 2015; Saurer et al., 2016; Taneja et al., 2014), owing to its wide range of applications including augmented reality (Middelberg et al., 2014), autonomous driving (Lim et al., 2012), adding and refining geo-tags in image collections (Hays and Efros, 2015; Zamir et al., 2014), large-scale 3D reconstruction (Crandall et al., 2011), and photo editing (Zhang et al., 2014).

Finding regions of interest has long been of great interest in computer vision. Much research has been done in the areas of feature selection, attention, and saliency (Hou and Zhang, 2007; Yoo et al., 2015; Mnih et al., 2014). Because task-relevant information is not generally uniformly distributed throughout an image (Almahairi et al., 2015), focusing on "interesting" areas, as opposed to "irrelevant" or even "distracting" areas, can often achieve better performance (Shrivastava et al., 2011; Kim et al., 2015; Girshick et al., 2016). This is especially true for image geo-localization, where challenges come not only from photometric and geometric changes between the query and the database images, but also from confusing visual elements (Knopp et al., 2010). For instance, features extracted from transient objects such as pedestrians and trees, or ubiquitous objects, like vehicles and fences, can introduce misleading cues into the geo-localization process.

To address this problem, there has been a recent push to intelligently select or reweight local features for image geo-localization (Arandjelović and Zisserman, 2014a; Schindler et al., 2007; Knopp et al., 2010; Kim et al., 2015). These methods focus on features with high distinctiveness in feature space (Arandjelović and Zisserman, 2014a) or in geographical space (Schindler et al., 2007; Knopp et al., 2010). Recently, Kim et al. (Kim et al., 2015) proposed

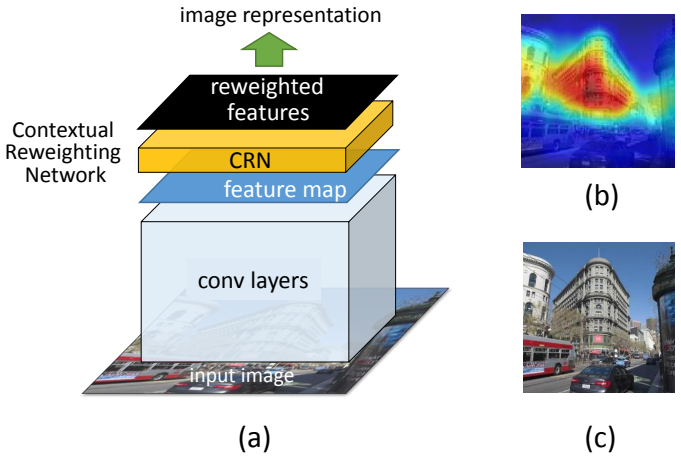


Figure 4.1: Image representation with contextual feature reweighting. (a) A contextual reweighting network takes convolutional features of a deep CNN as input to produce a spatial weighting mask (b) based on the learned contexts. The mask is used for weighted aggregation of input features to produce the representation of the input image (c).

a data-driven notion of "good" features as features that offer relatively high matching score to correct locations.

However, these approaches focus their analysis on individual local features in general. What is often overlooked is that a feature's usefulness depends largely on the context in the scene. For example, signage on buildings is useful for geo-localization, while signage on buses and t-shirts is misleading. There have been attempts to use top-down information such as semantic segmentation to restrict features to man-made structures (Mousavian and Košecka, 2015), or repetitive structure detection to avoid over-counting of visual words in the bag-of-words representation (Torii et al., 2015b). We point out that such supervised priors are limited and do not capture all relevant context about which regions to focus on for image geo-localization.

Hence, we aim to provide contextual guidance for reweighting features in a data-driven manner. To this end, we present a novel end-to-end convolutional neural network (CNN) model for learning image representations that adaptively reweight features based on the image context (Kim et al., 2017a). In particular, we introduce a *Contextual Reweighting Network* (CRN) that sits on top of the convolutional layers in a standard deep architecture

(Fig. 4.1 (a)). The CRN takes the feature maps of the base convolutional layers and estimates a weight for each feature based on its surrounding region (Fig. 4.1 (b)). By feature, we mean a column of activations at each spatial location of the feature maps. These weights are then applied to each feature as they are aggregated to produce an overall image representation.

We cast the image geo-localization problem as an image retrieval task and optimize the network with a triplet ranking loss based on the generated image representations. As a result, task-relevant contexts are discovered in an unsupervised manner, as the network learns in which context certain features should be emphasized or suppressed to better produce the spatial weighting, even though no ground-truth for the weighting nor the context information is provided. Visualizations of these learned contexts illustrate that the discovered contextual information contains rich high-level information that is not restricted to semantic cues, such as different types of buildings, vehicles, vegetation, and ground, but includes structural cues like lattice structure, different perspectives of buildings, and architectural styles.

Our training pipeline requires no training labels other than image geo-tags, which are commonly available on Internet photo collections such as Flickr. We use geometric verification to confirm the relationship between two views, then take the convex hull of matched inlier points to generate positive image pairs for training. In addition, we introduce efficient hard negative mining for image geo-localization, which can be seen as mimicking the image geo-localization process within a training batch.

To summarize, the innovations of this work are as follows: (1) We propose a novel end-to-end, fully-convolutional CNN for learning image representations that integrates context-aware feature reweighting. In particular, we introduce a contextual reweighting network that predicts weights for each region in the feature map based on its context. We experimentally validate that our pipeline significantly boosts the performance of the state-of-the-art methods. (2) We also show that unsupervised context discovery is achieved as a byproduct of training our network. The visualizations of these learned contexts illustrate that they capture rich high-level information. (3) We present a training pipeline where only commonly available

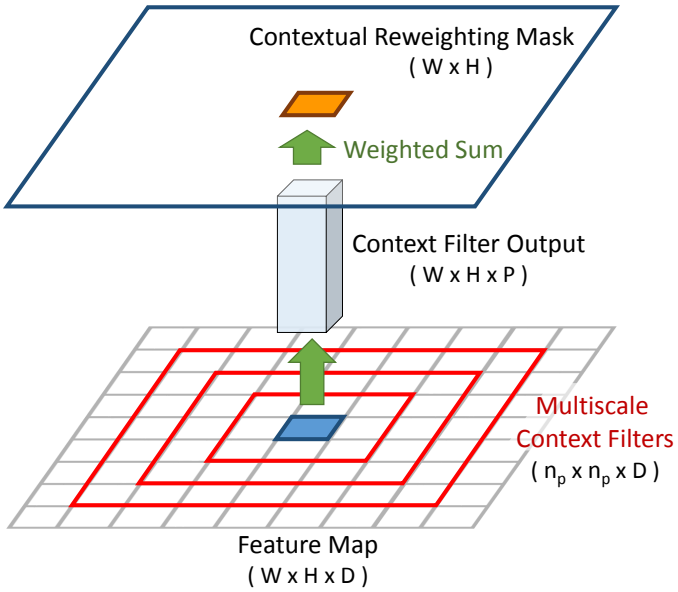


Figure 4.2: Contextual Reweighting Network. For each $1 \times 1 \times D$ convolutional feature, multi-scale contextual information is captured by P context filters with different window sizes ($n_p \times n_p \times D$). The filter output is then accumulated with learned weights to produce a reweighting value for the feature in question.

image geo-tags are required to automatically generate training data, with an efficient hard negative mining solution for image geo-localization.

4.1 Method

In this section, we first describe our model for learning image representations that integrate contextual reweighting of features (Sec. 4.1.1). We then illustrate our learning objective and the overall training process (Sec. 4.1.2).

4.1.1 The Contextual Reweighting Network

In order to integrate a learned contextual reweighting, we begin with a standard representation approach and add an auxiliary context reweighting network (CRN). This network takes features produced by the original representation as input and outputs a spatial weighting

over those features. The overall representation consists of a base network that generates mid-level features, a CRN, and a feature aggregation layer, as illustrated in Fig. 4.3.

Local Feature Representation: We treat activations of convolutional layers as local features. This has been shown to be effective for image geo-localization and image retrieval (Tolias et al., 2016b; Radenović et al., 2016; Azizpour et al., 2015; Arandjelović et al., 2016; Gordo et al., 2016). In particular, the $W \times H \times D$ dimensional feature maps of the last convolutional layer of the base network are treated as a set of D-dimensional local descriptors at $W \times H$ spatial locations. In our experiments, we used the conv5 output of AlexNet (Krizhevsky et al., 2012) and VGG16 (Simonyan and Zisserman, 2015)

Contextual Reweighting Mask: The CRN captures context information by using its hidden *context filters*, denoted as g_p, to explicitly look at $n_p \times n_p$ spatial windows around a local feature, as illustrated in Fig. 4.2. This is implemented using a convolution layer with kernel size n (Fig. 4.3). To obtain multi-scale contextual information, we use context filters with three different kernel sizes. These filters produce an activation map of $W \times H \times P$, where P is the total number of filters across scale.

The contextual reweighting mask m is computed as a weighted sum of these filter outputs, which is implemented with a $1 \times 1 \times P$ convolutional layer (Fig. 4.3) so that weights $\{w_p\}$ and the bias c are also learned during the training:

$$m = \sum_p w_p \cdot g_p(d) + c, \tag{4.1}$$

where d and $g_p(d)$ denote the feature maps and the output of filter g_p, respectively. Convolution layers in the CRN are followed by ReLU non-linearity. The resulting weighting mask is of size $W \times H$, and the values in the mask indicate which spatial regions of the feature maps are important.

As our model is fully convolutional, it is not restricted by image size. However, it is difficult to train context filters on different image resolutions. To bypass this issue, we scale the spatial range of the feature maps to a fixed scale (e.g., 13×13) before applying the

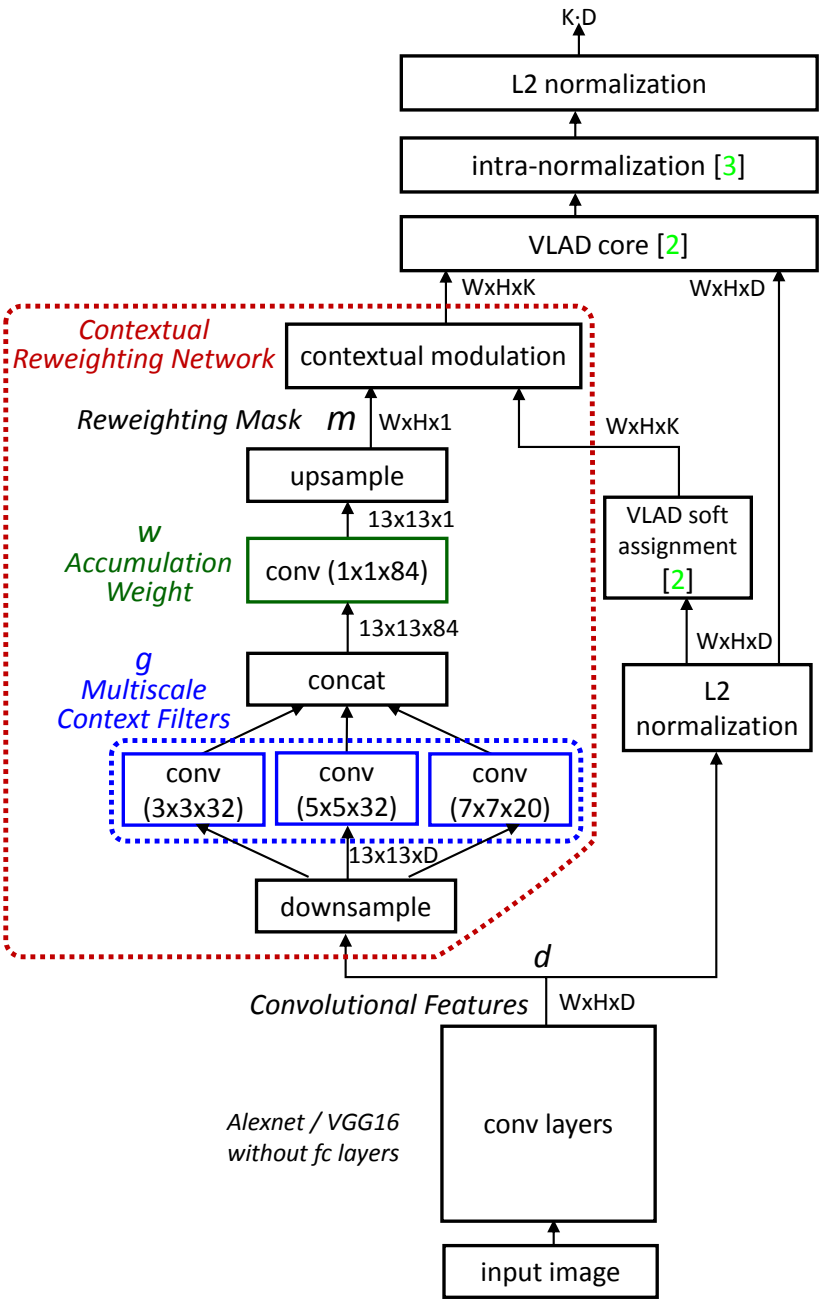


Figure 4.3: Overall network architecture. A CRN is a shallow network that takes the feature maps of convolutional layers as input and outputs a weighted mask indicating the importance of spatial regions in the feature maps. The resulting mask is used for performing context modulation for feature aggregation to create a global representation of the input image.

filters, and then rescale the computed contextual reweighting mask to the original feature map's spatial size. This provides image-level scale normalization, and is implemented with a matching downsampling (pooling)/upsampling layer pair which is applied at the beginning and end of the mask computation (Fig. 4.3).

Feature Aggregation via Contextual Modulation: Having a compact, fixed-length, global representation is necessary for efficient search and to limit memory requirements. We utilize the reweighting mask from our CRN to generate a fixed-length image representation f for the query and the geo-tagged reference images. The contextual modulation layer (Fig. 4.3) in the CRN adjusts the impact of a local feature d_l at spatial location l to the global representation f based on the reweighting mask value m_l at that location.

In our experiments, we use the trainable vector of locally aggregated descriptors (VLAD) layer (Arandjelović et al., 2016), which has been shown to be the state-of-the-art for place recognition and image retrieval tasks. The subvector of v that corresponds to the visual word k, denoted as v_k, is obtained as the accumulation of differences between local features d_l and the centroid c_k, weighted by the soft assignment a_l^k of d_l belonging to k, such that $v_k = \sum_{l \in R} a_l^k (d_l - c_k)$, where R denotes the set of spatial locations in the feature map.

Applying our context modulation, we obtain the reweighted VLAD representation f as follows:

$$f = \big[f_1, f_2, ..., f_K\big], \tag{4.2}$$

where

$$f_k = \sum_{l \in R} m_l \cdot a_l^k (d_l - c_k), \tag{4.3}$$

and K and R denote the number of visual words and a set of spatial locations in the feature map, respectively. This can be seen as a weighted pooling of features d_l. Following (Arandjelović et al., 2016), f is intra-normalized (Arandjelović and Zisserman, 2013), then L_2 normalized. The similarity between the two representations is computed as the inner product of the two. In this case, the contextual layer is implemented as a simple layer that takes m

and performs element-wise multiplication across all channels of the soft assignment output a ($W \times H \times K$).

Note that these local features, when learned end-to-end, inhibit activations on task-irrelevant visual elements based on the local appearance. On the other hand, our contextual modulation produces a spatially varying weighting based on semi-global context. Therefore, our overall model can be seen as combining both top-down and bottom-up guidance to determine on which areas to focus.

4.1.2 Training

Training Objective: In our setting, the geo-location of a query image I_q is approximated by finding the nearest neighbor reference images $\{I_r\}$ in feature space. Thus, the objective for learning our image representation f is to ensure that matching reference images I_r^+ are closer to the query image than non-matching ones I_r^-. To this end, we use a triplet ranking loss (Schroff et al., 2015; Wang et al., 2014; Gordo et al., 2016). During training, we provide image triplets to the network, each consisting of a training query image I_t, a positive reference image I_r^+, and a negative reference image I_r^-.

$$\begin{aligned} L_f(I_t, I_r^+, I_r^-) = \\ \max(0, \|f(I_t) - f(I_r^+)\|_2 - \|f(I_t) - f(I_r^-)\|_2 + \delta) \end{aligned} \tag{4.4}$$

By minimizing the triplet ranking loss (Eqn. (4.4)), the network learns in which context certain features should be emphasized or suppressed to better generate the spatial weighting mask m (Eqn. (4.1) and (5.4)), even though no ground-truth for the mask or the context is provided. Visualizations for the learned contexts and the masks are shown in Fig. 4.9 and Fig. 4.6 –4.7, respectively (Sec. 4.1.5).

Training Data Generation: We used images with GPS-tags as our training query image set $\{I_t\}$. We collected 6K Flickr images with GPS-tags, and 17K Google Streetview Research Dataset images, both covering the same region as the reference images in the evaluation

benchmark (Chen et al., 2011a). The images collected from Flickr are particularly challenging as they are unconstrained Internet photos that greatly vary from the reference images. Images from the Google Streetview Research Dataset are comparatively less challenging as they are also street view images taken from a vehicle, but they still differ significantly from reference images in terms of illumination, viewing angle, occlusion, and season. In order to further increase the size of our training data, we also added a randomly selected subset of the reference image set $\{I_r\}$ to $\{I_t\}$. Standard data augmentation techniques such as random cropping and re-lighting (Krizhevsky et al., 2012) were applied.

Given an image set with only GPS-tags, we want to automatically generate image triplets $\{I_t, I_r^+, I_r^-\}$ for training. To verify positive images, we use geometric verification. As in the previous chapter, for each training query image I_t, we define positive reference images I_r^+ as reference images that fall within 50m from the given GPS location and that pass geometric verification with respect to I_t by fitting a fundamental matrix (Hartley and Zisserman, 2003) using RANSAC (Fischler and Bolles, 1987) over SIFT (Lowe, 1999) matches. (Basic geometric transformations are described in Appendix B.) We select the top k_t images with the highest number of inliers. For I_r^-, we take the top k_t reference images that have the smallest distance to I_q based on the initialized image representation f (see implementation details) and are at least 225m away from the given GPS location, in order to mitigate possible geo-tag errors and lessen the effect of landmarks being visible from a large distance. The resulting I_r^+ and I_r^- are paired randomly to form k_t triplets $\{I_t, I_r^+, I_r^-\}$. In our experiments, k_t was set to 4.

To refine our training data, we perform ROI-based cropping and a scale test in order to account for $\{I_t, I_r^+\}$ pairs with small overlaps and large scale changes. A small overlap between I_t and I_r^+ could make the problem too difficult by giving the network misleading information that features on non-overlapping sides are not useful for image geo-localization. Therefore, we first perform a scale test using the area of the convex hull of the feature inliers from the geometric verification. Then we approximate the scale difference as the ratio of the areas of the two convex hulls. If they differ by more than a factor of 2, we exclude the triplet

from the training set. Also, if a pair passes the scale test, but the difference is more than a factor of 1.5, we crop and rescale the area around the center of the convex hull of I_t. In addition to making our training data more robust, this framework also expands training data by generating more training triplets. After the refinement, we ended up with 36K training query images each producing four triplets, giving a total of 144K training triplets for the San Francisco city area.

Hard Negative Mining: While previous work performed periodic full retrieval to update hard negatives (Arandjelović et al., 2016; Radenović et al., 2016), we present an efficient approach to mining hard negatives for image geo-localization. We mimic the image geo-localization process within the training batch. For every iteration, we perform image retrieval for I_q within the batch. We then select hard negatives I_r^- from the top retrieved images that are at least 225m away from the GPS location of I_t, similarly to how we selected initial I_r^-. As learning only based on the hardest negatives can lead to a bad local minima (Schroff et al., 2015), we also select some I_r^-'s randomly from the batch. In our experiments we used two each of the hard and the random negatives, and average their triplet losses and the corresponding gradients. We used the accumulation of gradients as a proxy for having a large batch size, averaging gradients of 25 batches.

Implementation Detail: Following (Arandjelović et al., 2016), we chose the number of centroids K in the VLAD representation to 64. The margin δ for the triplet ranking loss was set to 0.25. To initialize our model, we used Xavier random initialization (Glorot and Bengio, 2010) for our CRN. For the other layers, we used the parameters of NetVLAD models (Arandjelović et al., 2016) fine-tuned on our data using the same training procedure. In practice, we found that it was crucial to train the base convolutional layers, CRN, and VLAD layer jointly for convergence. We used a learning rate of 0.005 for the CRN, and 0.0005 for other layers except conv1, which we fixed to its pretrained state. We used a batch size of 24, and trained for approximately 10 epochs. For VGG16-based models, we fixed accumulation weights w_p to 1. We used images with a resolution of 480×480. For testing,

we averaged the image similarity computed by three patches (left, center, and right) similar to (Krizhevsky et al., 2012) for both our approach and NetVLAD, unless otherwise specified. Our implementation used Caffe (Jia et al., 2014).

4.1.3 Image Geo-Localization

Evaluation Dataset: For evaluation, we used the three standard benchmarks for image geo-localization. The first dataset is the San Francisco 1.1M benchmark dataset from the work of Chen et al. (2011a). It consists of query images taken with different mobile cameras in various settings, and reference images taken from vehicle-mounted wide-angle cameras. The ground-truth annotations for correct matches for each test query image are given in the benchmark.

We also evaluated our method on Tokyo 24/7 (Torii et al., 2015a) and Pittsburgh 250K test (Arandjelović et al., 2016), where a retrieved image is deemed to be correct if it is within 25m from the ground-truth position of the query. We used the corresponding training and validation sets for these datasets, namely, Tokyo Time Machine data (Arandjelović et al., 2016) and Pittsburgh 250K train/validation set (Arandjelović et al., 2016).

Evaluation Metric: We follow the evaluation protocol of (Torii et al., 2015b; Arandjelović and Zisserman, 2014a; Chen et al., 2011a; Baatz et al., 2012; Arandjelović et al., 2016; Tolias et al., 2016a), where performance is measured by the recall given the top N candidates in the shortlist. We also performed PCA whitening (learnt on the reference database for the San Francisco and on training images for Tokyo and Pittsburgh) on the obtained image representation $f(I)$ for both our method and NetVLAD (Arandjelović et al., 2016), reducing the dimension by half. In all of our experiments, we did not use post-processing such as geometric re-ranking.

Results on San Francisco 1.1M benchmark (Chen et al., 2011a)**:** To demonstrate the benefits of our context-aware image representation, we first compare our result to NetVLAD (Arandjelović et al., 2016). The only difference between our architecture and that of NetVLAD

is the existence of our proposed CRN that performs contextual feature reweighting. Fig. 4.4 depicts the recall curves of our method and NetVLAD based on AlexNet (Krizhevsky et al., 2012) and VGG16 (Simonyan and Zisserman, 2015). Our system with contextual feature reweighting consistently outperforms the systems without it. Our margin over NetVLAD at the top ($N = 1$) retrieved result is 4.8% for the AlexNet-based architecture, and 2.9% for the VGG16-based architecture. We implemented the VLAD layer proposed by Arandjelović et al. (Arandjelović et al., 2016) using Caffe (Jia et al., 2014) and used it in both our method and NetVLAD. Both networks were optimized in the same training pipeline (Sec. 4.1.2) for a fair comparison.

In Fig. 4.5, we compare our performance with other state-of-the-art methods. Our method achieves the best performance with a 83.2% recall at $N = 1$, exceeding the recall of the previous state-of-the-art (Tolias et al., 2016a) by 2.6%. The full comparison of recalls at the top of the shortlist ($N = 1$) with baseline methods is displayed in Table 4.1. The compared methods include binarized aggregated selective match kernel (ASMK*) (Tolias et al., 2016a), local distinctiveness based feature weighting (DisLoc) (Arandjelović and Zisserman, 2014a), repetitive feature re-weighting (Repttile) (Torii et al., 2015b), Hamming embedding (Jégou et al., 2010) with burstiness normalization (Jégou et al., 2009) (HE-BURST) (Jain et al., 2011), vocaburary tree with histogram equalization (NoGPS) (Chen et al., 2011a), and tf-idf weighting (Philbin et al., 2007). For ASMK*, DisLoc, Repttile, and NoGPS, we use the recall values reported by the authors. For HE-BURST, we used the recall reported in (Tolias et al., 2016a) using binary signatures of 128 bits. For tf-idf, we used the recalls reported in (Torii et al., 2015b).

We show examples of our results in Fig. 4.6 and 4.7, where the top retrieved images for each query image are displayed for our method and the NetVLAD (Arandjelović et al., 2016). As can be seen, our method retrieves correct reference images despite the existence of confusing objects, such as trees and cars (Fig. 4.6 (a)-(c),(f)), focuses its attention on

Method	% Correct
Ours (VGG16)	**83.2**
ASMK* (Tolias et al., 2016a)	80.6
NetVLAD (Arandjelović et al., 2016) fine-tuned (VGG16)	80.3
Ours (AlexNet)	**78.7**
DisLoc (Arandjelović and Zisserman, 2014a)	74.6
NetVLAD (Arandjelović et al., 2016) fine-tuned (AlexNet)	73.9
HE-BURST (Tolias et al., 2016a)	71.9
Repttile (Torii et al., 2015b)	65.4
NoGPS (Chen et al., 2011a)	41.2
tf-idf (Torii et al., 2015b)	23.2

Table 4.1: Proportion of correctly localized images at top 1

signage on stationary objects such as buildings (Fig. 4.6 (d)-(e) and Fig. 4.7 (c)-(d)), and distinguishes similar places with different details (Fig. 4.6 (e),(g) and Fig.4.7 (a)-(b), (e)-(g)).

To demonstrate the benefit of our learnt CRN, we also compared our method with Crow (Kalantidis et al., 2016) which performs feature reweighting in a predefined way. Crow creates a spatial weighting mask by computing the L_2 norms of the features at each spatial location, which results in emphasizing regions with high activations. Table 4.2 shows the performance of NetVLAD when CRN is replaced with CroW for spatial reweighting, all of which underperform our proposed CRN.

We further compare the performance of using different image resolutions in Table 4.3. The networks are independently trained for each different resolutions of images, but with the same architecture, as well as the same size of receptive fields. The recall decreased for both our method and NetVLAD when the image resolution was reduced to 321×321. This indicates that the fine details in large resolution images are important for retrieving the same instance of a place. Also, we see that performance of NetVLAD decreased although a larger context is taken into account for the lower resolution images, which implies that using larger context by itself does not necessarily improve the performance. Our method, on the other hand, makes use of higher-level information as a guide for where to focus on, but still aggregates lower-level features for the image representation for preserving fine details.

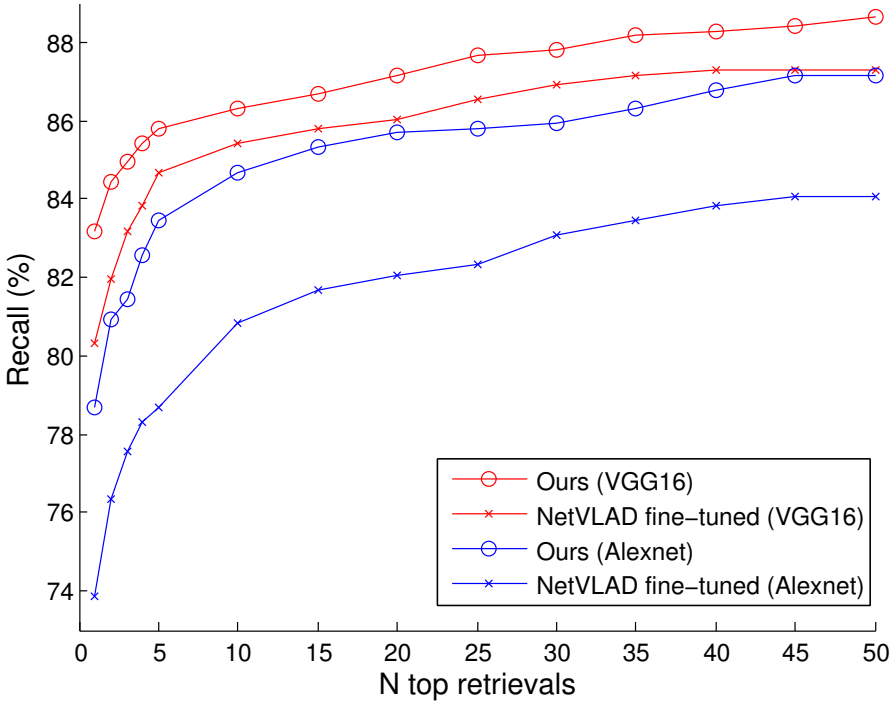


Figure 4.4: Recalls with and without contextual feature reweighting.

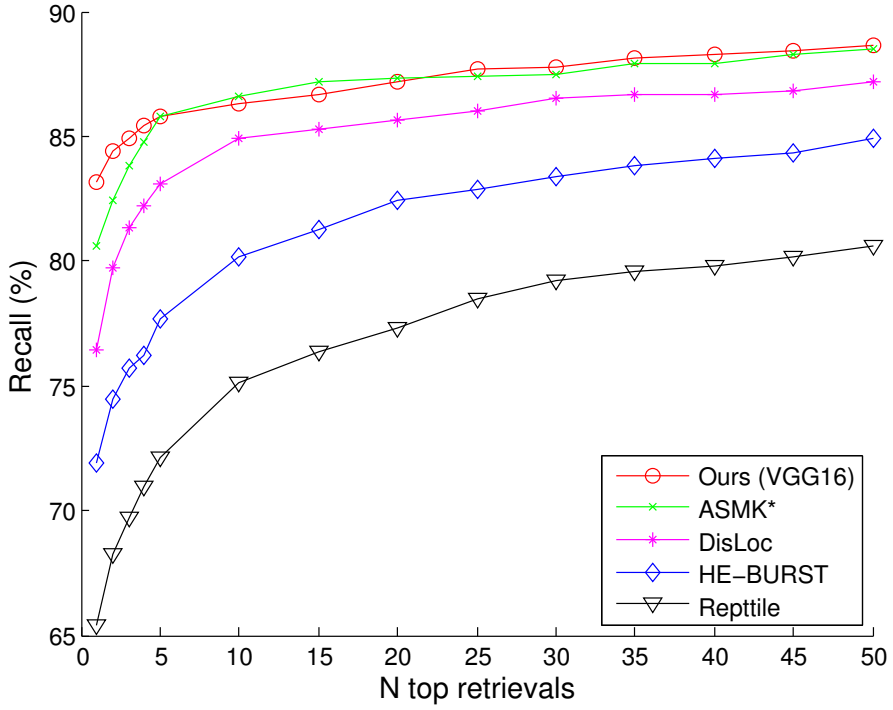


Figure 4.5: Comparison of recalls with the state-of-the-arts methods.

Table 4.2: Comparison of our proposed CRN and CroW (Kalantidis et al., 2016) with (V)GG16 and (A)lexnet base architectures.

Methods	top-1	top-5	top-10	top-25
CRN+NetVLAD (V)	**83.2**	**85.8**	**86.3**	**87.7**
CroW+NetVLAD (V)	80.1	84.3	85.3	86.5
CRN+NetVLAD (A)	**78.7**	**83.4**	**84.7**	**85.8**
CroW+NetVLAD (A)	74.1	79.3	80.8	82.3

Table 4.3: Comparison of using different image resolutions. All models are based on AlexNet architecture.

Resolution	Methods	top-1	top-5	top-10
480 × 480	**CRN+NetVLAD**	**78.7**	**83.4**	**84.7**
	NetVLAD	73.9	78.7	80.8
321 × 321	**CRN+NetVLAD**	**76.6**	**81.7**	**82.4**
	NetVLAD	69.9	75.0	76.4

Results on Tokyo 24/7 (Torii et al., 2015a) **and Pittsburgh 250K test** (Arandjelović et al., 2016)**:** Table 4.4 displays the results of our method evaluated on the Tokyo 24/7 and Pittsburgh 250K test datasets. We used the same training and validation sets as in (Arandjelović et al., 2016), but with our training pipeline (Sec. 4.1.2). Our method consistently outperforms the state-of-the-art NetVLAD (Arandjelović et al., 2016) on Tokyo 24/7, with a margin of 3.4% for all test images and 5.3% for the challenging sunset/night-time images at $N = 1$. Our performance on Pittsburgh 250K test data is similar to that of NetVLAD. We suspect this is due to lower variablity between the query and the reference images, in which case it may not be beneficial to down weight certain features. Also, while the query images for Pittsburgh 250K consist of randomly sampled streetviews, our training pipeline may have introduced a bias to the network by dropping training query images that do not pass geometric verification, utilizing only 72% of the training query set.

Table 4.4: Recalls on Tokyo 24/7 (Torii et al., 2015a) and Pittsburgh 250k test (Arandjelović et al., 2016) datasets. All models are based on VGG16 architecture. For NetVLAD, we used the recalls reported by authors of (Arandjelović et al., 2016). We used the full resolution images for evaluation as in (Arandjelović et al., 2016).

data	set	method	top-1	top-5	top-10
Tokyo 24/7 (Torii et al., 2015a)	all	**Ours**	**75.2**	**83.8**	**87.3**
		NetVLAD	71.8	82.5	86.4
	sunset /night	**Ours**	**66.7**	**76.7**	**81.9**
		NetVLAD	61.4	75.7	81.0
Pittsburgh 250K (Torii et al., 2015b)	test (Arandjelović et al., 2016)	**Ours**	85.5	**93.5**	**95.5**
		NetVLAD	**86.0**	93.2	95.1

4.1.4 Comparison of the Emphasized Features

We qualitatively compare the emphasis on the features of our context-aware image representation and NetVLAD (Arandjelović et al., 2016) in Fig. 4.8. We visualize the weighted mask generated by CRN for our method. To measure which regions of the feature maps were emphasized for NetVLAD, we computed the change in representation in Euclidean distance when leaving out each 1×1 spatial window in the convolutional feature maps. As can be seen, our method focuses on regions that are useful for image geo-localization while avoiding confusing visual elements. Moreover, it is capable of emphasizing the distinctive details on buildings. On the other hand, the NetVLAD (Arandjelović et al., 2016) is inherently limited as it emphasizes local features independently; many features on confusing scene elements such as vegetation, pedestrians, and vehicles are emphasized.

4.1.5 Unsupervised Discovery of Contexts for Image Geo-Localization

To visualize the learned contexts, we display the image patches with the highest responses. That is, for each learned context filter g_p, we collect the strongest responses in each sampled image from the database. We crop out the square image patch from the original image at the center of the scaled feature map with the width of $n_p \frac{W_I}{W}$, where n_p, W_I, and W are kernel size of g_p, image width, and feature map width, respectively. Although the network is optimized

Figure 4.6: Example retrieval results on San Francisco benchmark dataset. From left to right: query image, our contextual reweighting mask in heat map, the top retrieved image using our method, the top retrieved image using NetVLAD (Arandjelović et al., 2016). Green and red borders indicate correct and incorrect retrieved results, respectively. Results are based on our AlexNet-based model.

Figure 4.7: Example retrieval results on San Francisco benchmark dataset. From left to right: query image, our contextual reweighting mask in heat map, the top retrieved image using our method, the top retrieved image using NetVLAD (Arandjelović et al., 2016). Green and red borders indicate correct and incorrect retrieved results, respectively. Results are based on our AlexNet-based model.

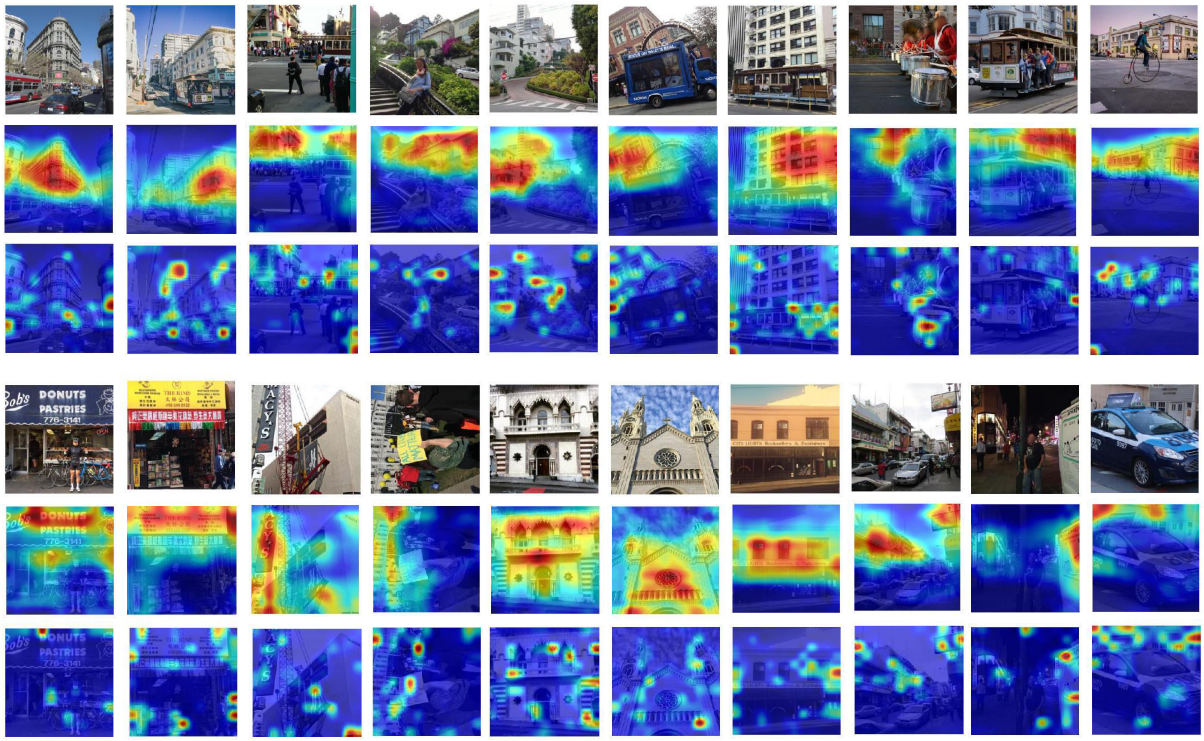


Figure 4.8: Comparison of emphasis on features. (first rows of heat maps below images) Our contextual reweighting mask. (second rows of heat maps below images) NetVLAD (Arandjelović et al., 2016) emphasis on features. Both models are based on AlexNet architecture.

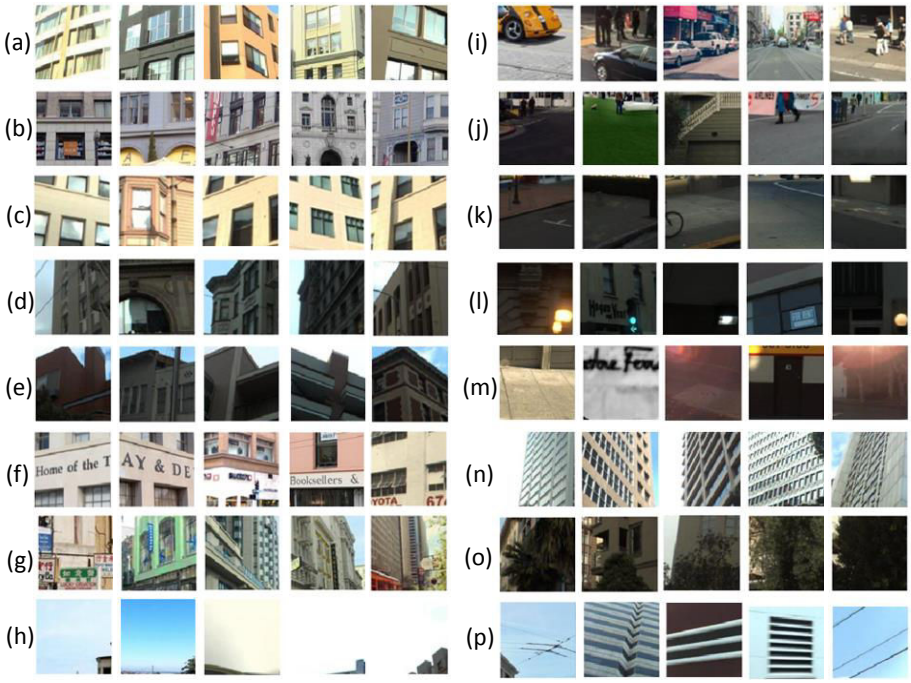


Figure 4.9: Discovered data-driven contexts for image geo-localization. For each learned context filters g_p, we display image patches with top responses (Sec. 4.1.5). (Left) Filters assigned positive weights $w_p > 0$. (Right) Filters assigned negative weights $w_p < 0$. Results are based on our AlexNet-based model.

only under the loss for image geo-localization, we observe that interesting contexts were captured by our contextual filters through the learning process. The results are shown in Fig. 4.9, where visualization of the context filters are aligned based on the sign of the accumulation weights w_p. If w_p is positive, it means the context captured contributes to assigning positive weights on the feature in question. The reverse is true for negative w_p. These contexts not only highlight semantic cues like buildings, vehicles, and pedestrians (Fig. 4.9 (a),(i-k),(o)), but also structural information such as the geometric changes in buildings (Fig. 4.9 (d-e)), sky lines (Fig. 4.9 (h)), architectural styles (Fig. 4.9 (b-c)), and buildings with signs (Fig. 4.9 (f-g)). Notably, even without supervision, our model assigns negative accumulation weights to lattice structures (Fig. 4.9 (n)), which is similar to what has previously been achieved with supervision (Torii et al., 2015b) that lowers weights for features occurring in repetitive structures.

Table 4.5: Retrieval performance of our model trained on San Francisco on image retrieval benchmarks Oxford 5K and 105K (Philbin et al., 2007). No cropping of ROI in the query, spatial re-ranking, or query expansion was performed. The accuracy is measured by the mean Average Precision (mAP). All compared models are based on VGG16 architecture.

	Oxford 5K			Oxford 105K	
Method	**Ours**	NetVLAD (Arandjelović et al., 2016)		**Ours**	NetVLAD (Arandjelović et al., 2016)
Train / Dim	SF	SF	PGH	SF	SF
16384	**0.704**	0.683	-	**0.685**	0.664
8192	**0.699**	0.682	-	**0.680**	0.660
4096	**0.692**	0.672	0.691	**0.671**	0.651
2048	**0.683**	0.660	0.677	**0.662**	0.633
1024	0.667	0.650	**0.669**	**0.644**	0.625
512	0.645	0.626	**0.656**	**0.622**	0.598
256	**0.642**	0.608	0.625	**0.617**	0.579
128	**0.615**	0.569	0.604	**0.586**	0.540

4.1.6 Image Retrieval

To assess generalizability of our approach, we evaluated our image representation using the CRN, trained on San Francisco, on standard image retrieval benchmarks (Philbin et al., 2007) without any fine-tuning. The results are shown in Table 4.5. We compared with NetVLAD (Arandjelović et al., 2016) trained in the same pipeline as ours on San Francisco (SF), and the one that is trained on Pittsburgh (PGH) as reported in (Arandjelović et al., 2016). For all methods, we did not perform cropping of the ROI in the query, spatial re-ranking, or query expansion. Our model outperforms both representations. Especially, it consistently exceeds the mAP of NetVLAD trained on the same dataset by 2-4% margins.

4.2 Contextual Feature Reweighting

To show how our Contextual Reweighting Network (CRN) adaptively weights features based on the context, we generated contextual reweighting masks on synthetic images. Figure 4.10 (left) shows an example where features on the signage on a store front are assigned a comparably high weight by our CRN within the context of its original image. It can be

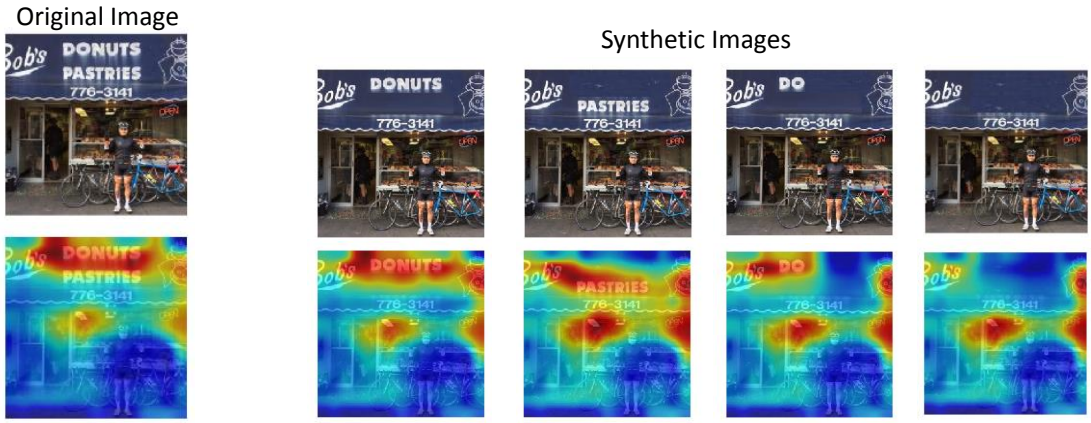


Figure 4.10: High weights are assigned on features from the signage on a store front. As we removed the letters on the signage, the weights diminish. (top) Input image. (bottom) Generated contextual reweighting mask in a heat map (red: high, blue: low).

observed that the high weighting is caused by the signage as the weight diminishes when removing some of the letters on the signage as shown in Figure 4.10 (right). To see how CRN changes weights on a feature from one context to another, we cropped out some image patches containing the letters on the signage on Figure 4.10 and placed them on other images such that they are surrounded by different visual elements such as pedestrians (Figure 4.11 (a-b,d)), vehicles (Figure 4.11 (c,h)), vegetation (Figure 4.11 (e)), and sky (Figure 4.11 (f)). We directly overlayed the image patch without resizing. It can be observed that the letters on the signage are no longer assigned high weights as their surroundings changed. For Figure 4.11 (i-j), we pasted the store signs that are from the same image. The results are generated from our AlexNet (Krizhevsky et al., 2012)-based model.

4.3 Retrieval Result: Geo-Localization

More examples of retrieval results on the San Francisco benchmark (Chen et al., 2011a) for image geo-localization using our context-aware image representation are depicted in Figure 4.12 and Figure 4.13, each using our AlexNet (Krizhevsky et al., 2012) and VGG16 (Simonyan and Zisserman, 2015) based models. The top 5 retrieved images are shown for

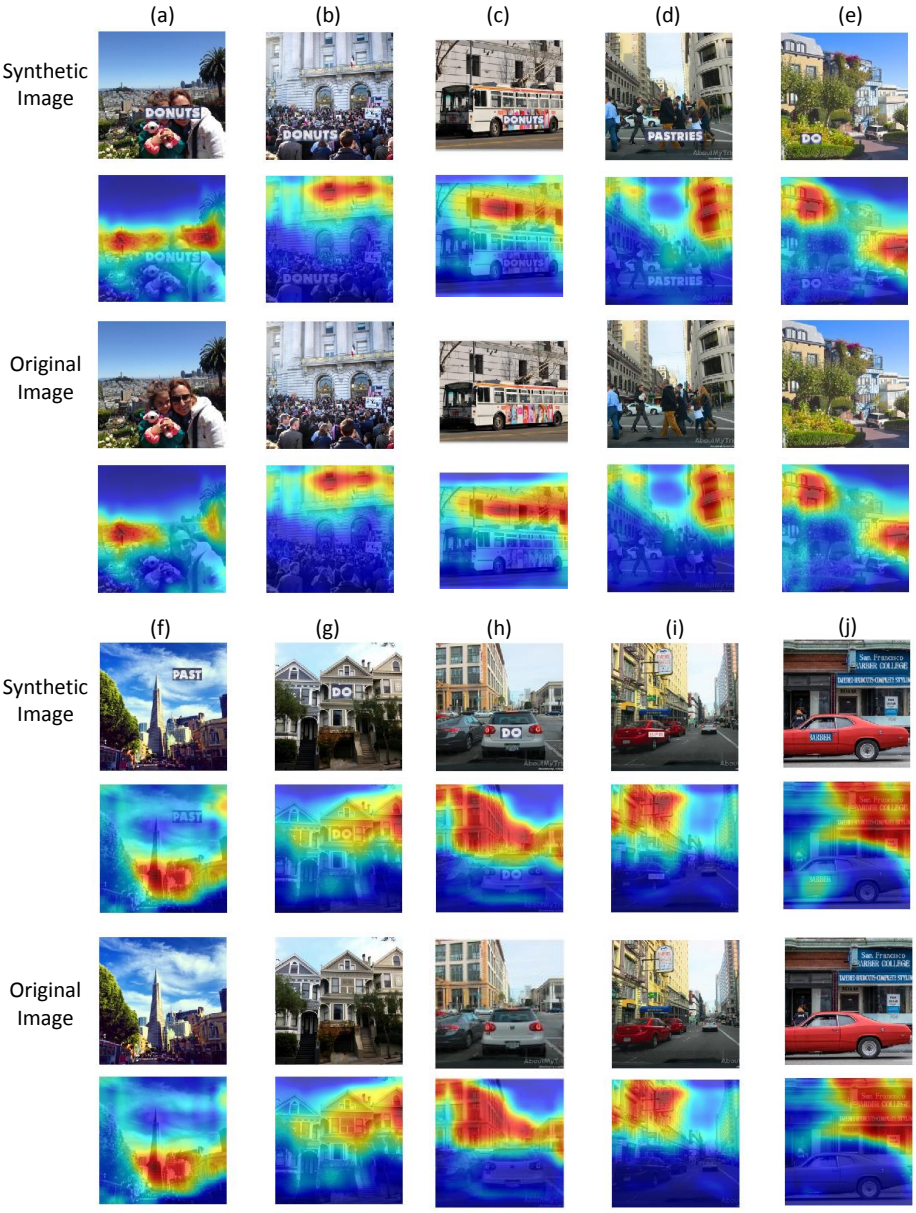


Figure 4.11: We generated synthetic images by pasting image patches containing the letters of the signage from Figure 4.10 that was assigned high weights at the store front (a-h). For (i)-(j), we overlayed the store signages from the same image on vehicles. Generated contextual reweighting masks are visualized on the bottom of each image as a heat map (red: high, blue: low). The letters from the signagage are no longer assigned high weights as the surrounding contexts have changed.

each query image. The results of NetVLAD (Arandjelović and Zisserman, 2014a) trained on San Francisco in the same pipeline as ours (with the same base architecture) are also shown for comparison.

4.4 Conclusions

In this chapter, we introduced a novel Contextual Reweighting Network that learns image representations incorporating context-aware feature importance. We demonstrated that our CRN-based representation improves upon the existing state-of-the-art accuracy for geo-localization. The visualization of the outputs of our CRN shows that the relevant context for image geo-localization is captured as a byproduct of training our network. We also provide an efficient training pipeline using only images with geo-tags. Our proposed CRN can be combined with other feature aggregation methods, and can be applied to other problems such as object recognition.

Figure 4.12: Image geo-localization results. (left) Query images and the corresponding contextual reweighting masks generated by our CRN as heat maps, (right) Top five retrieved images using our method and NetVLAD (Arandjelović and Zisserman, 2014a). The green boxes around the retrieved images denote the correct results. The results are based on our AlexNet-based model.

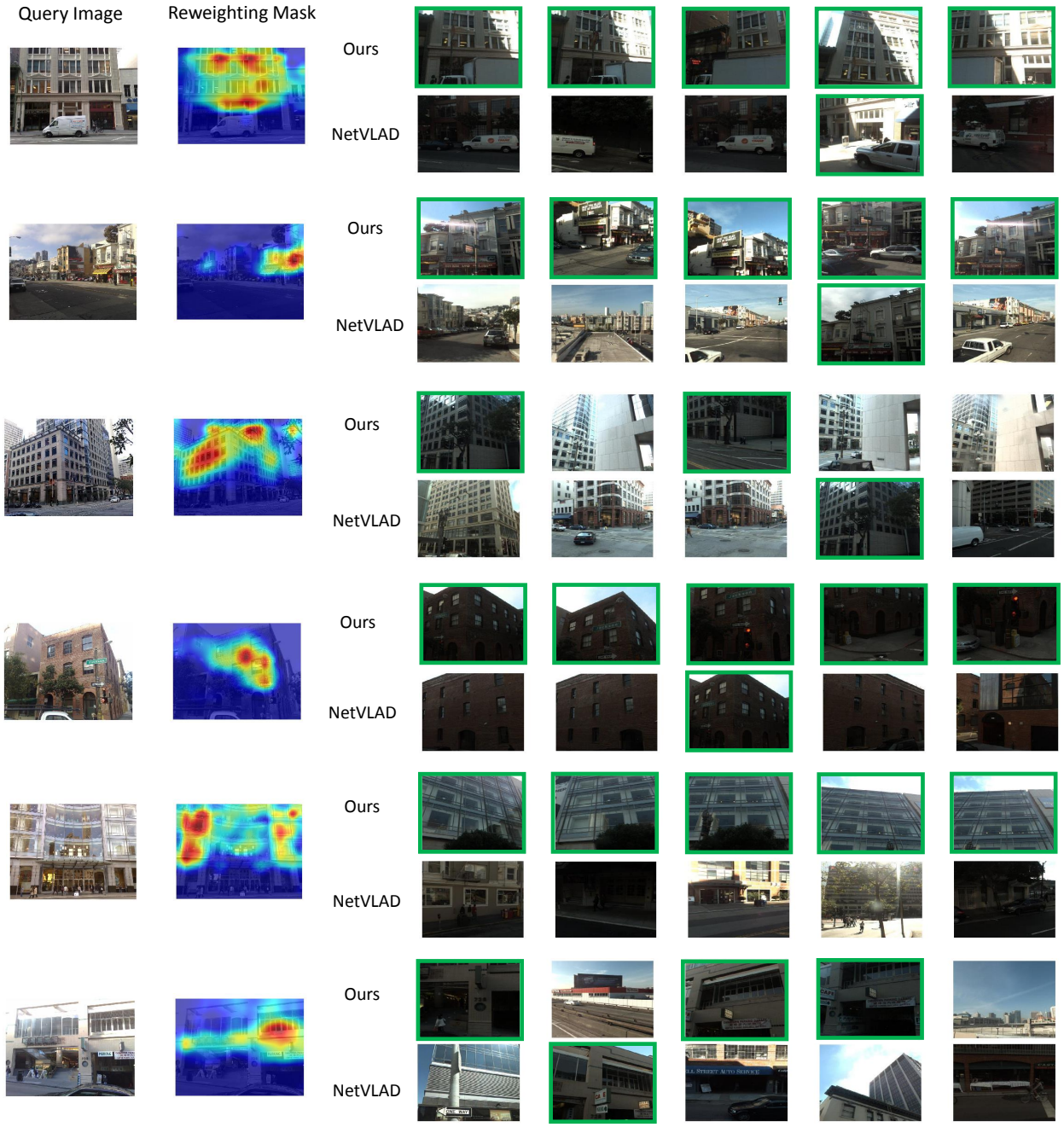


Figure 4.13: Image geo-localization results. (left) Query images and the corresponding contextual reweighting masks generated by our CRN as heat maps, (right) Top five retrieved images using our method and NetVLAD (Arandjelović and Zisserman, 2014a). The green boxes around the retrieved images denote the correct results. The results are based on our VGG16-based model.

CHAPTER 5: HIERARCHY OF ALTERNATING SPECIALISTS FOR SCENE RECOGNITION

Accurately identifying the background in an image (*e.g.* beach, mountains, candy store) is an important task in computer vision because it provides us with strong contextual information as to what is happening in the scene. Despite the recent progress in the visual category recognition, we are still short of scene recognition systems that are competitive with human-level performance. The core challenge that needs to be addressed is the severe intra-class variation and inter-class similarity. Not only there are many visually diverse instances within one scene category (*e.g.* Waikiki Beach vs. Praia do Castelo), but there is also a significant visual overlap between different scene categories (*e.g.* airports with shopping areas vs. indoor shopping centers).

Several approaches have been proposed to address this problem by designing or learning better visual features (Wang et al., 2012; Zheng et al., 2011; Somanath and Kambhamettu, 2010; Guo et al., 2010; Cheng et al., 2016; Chen et al., 2017a). Newer end-to-end deep neural networks were able to ease such efforts, achieving state-of-the-art classification accuracy (Agrawal et al., 2014; Zhou et al., 2014b). However, it becomes increasingly hard to find a distinctive representation when the classes become visually nearly indistinguishable as the number of classes increases (Qian et al., 2015). Downweighting the representations for commonly shared visual elements can help reduce the inter-class similarity. However, these elements are sometimes key to distinguishing a class from remaining other classes, as illustrated in Fig. 5.2.

Thus, a sensible way to handle this issue is to apply a divide and conquer (Tu, 2005) strategy to dedicate different CNNs to separable subproblems. Existing methods organize classes into coarse categories, either based on the semantic hierarchy (Goo et al., 2016; Zhao

et al., 2011; Hwang and Sigal, 2014; Deng et al., 2014) or the confusion matrix of a trained classifier (Murthy et al., 2016; Yan et al., 2015; Warde-Farley et al., 2014). However, we observe that there are multiple modes of intra-class appearance variation, and that each of these modes typically causes overlap with different subsets of categories. As depicted in Fig. 5.3, some images of a kitchen with cabinets can be confused with a bathroom or a bedroom with similar furnishings, while other kitchen images showing the dining area are easily mistaken as a bar or a restaurant. In this case, grouping the whole kitchen class with the whole bathroom or restaurant class into a coarse category is suboptimal. Instead, it would be more effective to group confusable images below the category level, such as the images of different classes with similar furnishings as shown in as shown in Fig. 5.3.

Hence, we aim identify such *confusing clusters* of images in a coarse to fine manner based on high-level appearance. The key idea is to disentangle intra-class variation and inter-class similarity by limiting the intra-class variation within each cluster. With reduced intra-class variation, a specialist model can focus on finding the subtle differences between the categories within the cluster. To this end, we introduce a Hierarchy of Alternating Specialists model, which automatically builds a hierarchical network of specialists based on the unsupervised discovery of confusing clusters (Kim and Frahm, 2018). For a given specialist CNN, we find its corresponding confusing cluster by performing clustering in the feature space of its parent model that handles a more general task. This groups images that are visually similar and likely to be confused by the parent model. For assigning images to a model in the hierarchy, we propose a simple routing function inspired by the ratio test (Lowe, 1999), which invokes only a small fraction of the models in the whole tree for an input image. Our results show that our approach achieves better performance than the conventional approach of using coarse categories.

On the other hand, we notice that the spatial layout and the objects in the scene are complimentary features for scene categorization. This seems natural because the scene class is often determined by the way humans use these objects in this spatial context. For example,

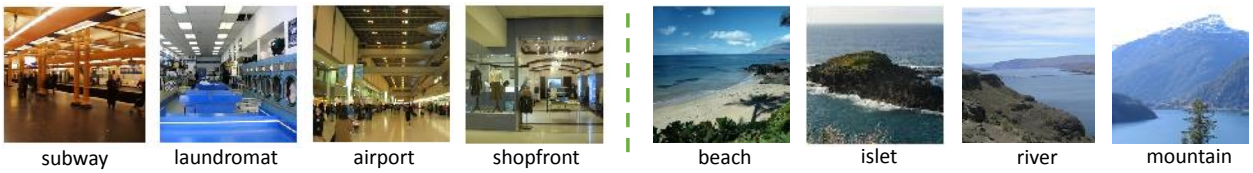


Figure 5.1: (left) Similar layouts make these scenes confusing, but different objects within the scene can help determining the correct scene class. (right) While these scenes are similar in terms of content, the layout of the scene can help distinguish between them.

the different rooms in a house are typically similar in structure with walls, doors, and windows. However, the objects, such as furnishings of the rooms determine their function as being a living room, office, or dinning room. Another notable fact is that the objects do not necessarily stay in the same configuration such as the chairs in a dinning room or people in an airport. To account for this fact, we use two different types of representations in our architecture. One with global order-less pooling of local activations to account for transient local visual elements (Zhou et al., 2014a), and the other with global ordered pooling that preserves spatial information (Gong et al., 2014). In particular, we propose an alternating architecture, where the architecture of a specialist alternates between the two representations based on its level in the hierarchy. We show in the experiments that our approach achieves better performance than both the fused features and the hierarchical architecture with a single type of representation.

In summary, our innovations are as follows: (1) We propose a hierarchical generalist-specialist model that automatically builds itself based on the unsupervised discovery of confusing clusters in a coarse to fine manner. The confusing clusters allow specialists to focus on subtle differences between images that are visually similar and confusable to their parents. We experimentally validate that our method significantly outperforms baselines including tree-structured models based on coarse categories. (2) We propose a novel alternating architecture that effectively takes advantage of two complementary representations that capture spatial layouts and transient objects. As minor innovations, we introduce a novel routing function as well as mini-batch soft k-means for end-to-end fine tuning. Beyond the detailed innovations,

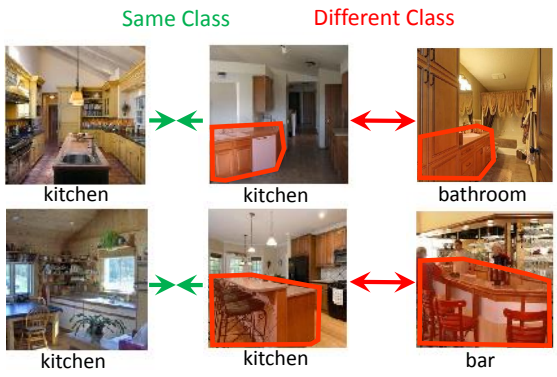


Figure 5.2: Examples of intra-class variation and inter-class similarity. While base cabinets and bars characterize the kitchen class, it causes overlap with other classes at the same time.

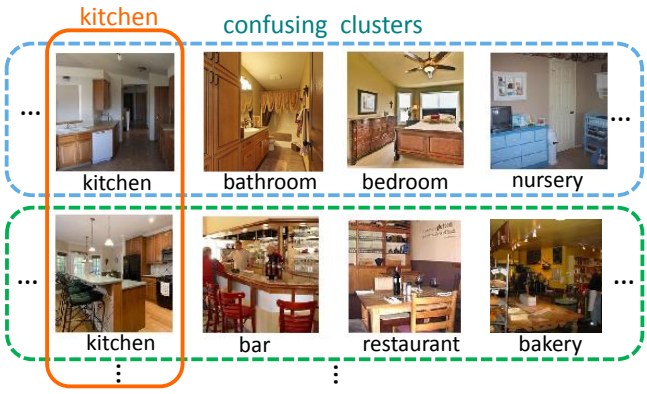


Figure 5.3: There are subsets of images in each class that are often confused with those of other classes. We discover *confusing clusters* in the feature space to disentangle intra-class variation and inter-class similarity.

our proposed algorithm is generalizable to other categorization tasks, and is applicable to any CNN architecture.

5.1 Method

In this section, we first describe our proposed hierarchy of specialists with alternating architectures that composed of two complementary feature representations in Sec. 5.1.1.

We then illustrate how to discover a specialist's area of expertise in an unsupervised manner in Sec. 5.1.2. Lastly, we describe the learning objectives as well as the overall training procedure in Sec. 5.1.3.

5.1.1 Hierarchy of Alternating Specialists

We propose a hierarchical version of the generalist-specialist models (Hinton et al., 2015), where the child specialist focuses on the task that is more specific than its parents. To achieve this, we begin with a generalist model and then incrementally add specialist models in the next level of the hierarchy, after reaching convergence at the current level. We initialize a new specialist with its parent, or the nearest ancestor that share the network architecture, to

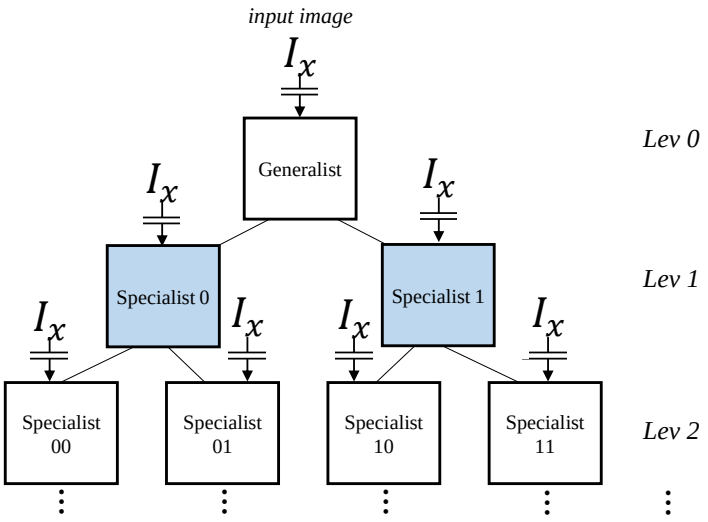


Figure 5.4: Our proposed hierarchy of alternating specialists, where a child model focuses on the task that is more specific than its parent. The assignment to a specialist is determined by our novel routing function, depicted as switches. The white and the blue shaded box denote network architectures with different global pooling strategy.

inherit its parent's knowledge as they encode important commonalities of the classes. Note that a specialist model outputs predictions for the same set of categories as the generalist model does. A specialist refines the inherited model towards the finer details to distinguish the classes for images that fall into its specialty. The overall architecture is depicted in Fig. 5.4. Similar to (Murthy et al., 2016), the algorithm stops extending the hierarchy when there is no further improvement during validation, or if the network reaches a pre-specified maximum depth. In this paper, we use a binary tree structure where each parent model has two child models. Every model within this tree shares the low level layers for computational efficiency.

We design this hierarchy of specialists to have an alternating architecture such that specialists at each level have a different model architecture than their parents or children. In particular, we use the global ordered pooling architecture for capturing the rough geometry of the scene, and the global orderless pooling architecture for capturing transient visual elements such as objects. The key idea is that the scene layout and the objects in the scene are complementary for scene classification. Objects can often disambiguate the two

images belonging to different categories with similar layouts, while the scene layouts can help distinguish two images, which share the same objects (Fig. 5.1).

The two architectures differ from each other in how they pool the features in the last convolutional layer before the fully connected layers for the class prediction. First is the global ordered pooling architecture, where the orderless pooling operation (i.e., max- or average-pooling) is performed only within a local spatial window. Thus, the representation preserves the coarse spatial information. This type of architecture is found in networks such as Alexnet (Krizhevsky et al., 2012) and VGG (Simonyan and Zisserman, 2015). The second is the global orderless pooling architecture, in which convolutional features are pooled through global average-pooling, global max-pooling, or VLAD (Arandjelović et al., 2016). This has a high degree of invariance for encoding local visual elements such as objects, analogous to the widely adopted bag-of-words representation. Examples of this type of architecture include NIN (Lin et al., 2013) and ResNet (He et al., 2016).

Our model uses the original pooling strategy of the base architecture for the generalist at the root node, and alternates between the two architectures for all other elements of our tree structure. To convert one architecture from the other, we either drop global orderless pooling (global orderless $\rightarrow$ global ordered) and substitute it with a fully-connected layer, or we replace the fully connected layer with global average pooling (global orderless $\leftarrow$ global ordered).

Routing: In order to decide which model in the hierarchy should tackle the input image, we use a simple routing function inspired by the SIFT ratio test (Lowe, 1999). The idea is to let the parent (generalist) handle the image unless the image has a good membership to any of its children (specialist)'s area of expertise. We define the routing function to produce a k-dimensional binary vector γ, where the k is the number of children at the current node and $\sum_i \gamma_i \leq 1$. $\gamma_i = 1$ indicates if the routing to the i-th child is valid. In the feature space of the parent f_g, given its childs' corresponding cluster centroids μ_k's, we compute the distance between the input image I and its nearest centroid μ_i, where $i = \underset{k}{\operatorname{argmin}} ||f_g(I) - \mu_k||$. We

also compute the second nearest centroid μ_j. We then take the ratio of the two distances. If the ratio is less than a threshold τ, the image is assigned to specialist i. Otherwise, the image is assigned to the generalist at the current node (Eqn. (5.1)). Then the same routing procedure is performed at node i. The decision boundary of this routing function consists of two Apollonius circles where the foci's are the centroids μ_i and μ_j (Aljundi et al., 2017).

$$\gamma_i{}^{train}(I) = \begin{cases} 1, & \frac{||f_g(I)-\mu_i||}{||f_g(I)-\mu_j||} < \tau \\ 0, & otherwise \end{cases} \tag{5.1}$$

During testing, we put an additional constraint for selecting the specialist based on the relative confidence of the prediction between the specialist and the generalist. Intuitively, for those images that are within the area of the specialist ($\gamma_i{}^{train}(x) = 1$), we trust the prediction of the specialist as our answer, when the confidence of the specialist is greater than that of the generalist on the given image. Otherwise, we accept the generalist's prediction and regard the prediction of the specialist as unreliable.

$$\gamma_i{}^{test}(I) = \begin{cases} 1, & \gamma_i{}^{train}(I) \ \wedge \ (conf_i(I) > conf_{gen}(I))) \\ 0, & otherwise \end{cases}, \tag{5.2}$$

where $conf_t(I) = \max_c P(c|I, \theta_t)$. Since the distance to the clusters is computed in the feature space of the parent models at each level, the total number of models that needs to be invoked is $n_l + 1$ where n_l is the hierarchical level of the selected model ($n_l = 0$ for the generalist). The procedure can also be computed in parallel, at the expense of the number of invoked models (see Sec. 5.2.8).

5.1.2 Discovering the areas of confusion

We want to partition the input data based on their feature representation, and not by their categorization, thus allowing samples belonging to the same class to fall into different

clusters. Our key insight is that each subset within a class is often associated with different types of inter-class similarity. We perform clustering in the feature space of a more general classification model to discover *confusing clusters*, the groups of images that are both visually similar and likely to be confused with each other in an unsupervised manner. This can be interpreted as disentangling intra-class variation and inter-class similarity, as the resulting cluster has limited intra-class variation, and a specialist model can focus on finding the subtle differences between each categories within the cluster. Also, due to our alternating architecture, we obtain confusing clusters that are both confusing in terms of scene layout and the transient scene objects as we go deeper in the hierarchy.

Feature for clustering: The penultimate layer of a generalist encodes high-level appearance information and is fed to a linear classifier. On the other hand, the features from the last fully-connected layer directly encode the class scores by the generalist. The distance of images in these two embedding spaces indicates how likely they are to be distinguished by the generalist. In the dataset we tested, the combination of these embeddings produced a marginally better results compared to using each of them separately. In the experiments, we report the result using the combined features, unless otherwise specified.

Incremental hard clustering: In the previous subsection, we have built our hierarchical model in an incremental manner, where the models in the next hierarchical level are added when their parent models have converged. As such, we discover confusing clusters by performing hard k-means clustering on the features of a converged parent model. Once initialized with these clusters, we can further fine-tune them end-to-end using the soft k-means layer described below.

Soft k-means layer for fine-tuning: We propose to use a mini-batch-based soft k-means that allows end-to-end fine-tuning. For each model θ, we update the centroids μ_k through back-propagation to optimize the following soft k-means objective function:

$$L_{clust}(\theta, \mu; I_i) = \sum_{k=1}^{K} \sum_{i=1}^{N} w_{ik} \| f_\theta(I_i) - \mu_k \|^2, \tag{5.3}$$

where

$$w_{ik} = \frac{e^{-m\|f_\theta(I_i)-\mu_k\|^2}}{\sum_{k=1}^{K} e^{-m\|f_\theta(I_i)-\mu_k\|^2}}, \tag{5.4}$$

and $f_\theta(I_i)$ denotes a image representation in the mini-batch. The parameter m decides the softness of the membership w_{ik} of x_i belonging to cluster k. We set m to $1/(8\sigma^2)$ where σ is the average of the standard deviation to the cluster center, which is computed during the hard k-means clustering.

5.1.3 Training

5.1.3.1 Classification Loss

As we allow the samples belonging to the same class to be in different clusters, it may introduce class imbalance in the training set of the specialists. Thus, we weigh the cross-entropy loss with the inverted document frequency similar to (Lin et al., 2017). This better accounts for under-represented classes within the cluster. We computed the inverted document frequency as a running average to allow changes caused by clustering.

$$L_{class}(\theta; x) = -\sum_{c} \Big(\log \frac{N}{n_c} \Big) \log(P[c|x, \theta]) \tag{5.5}$$

Training Objective: Our final training objective consists of clustering loss and classification loss as follows:

$$L_{total}(\theta; x) = \sum_{d \in \mathcal{D}} L_{class}(\theta_d; x) + \sum_{d \in \mathcal{D}, d \notin \mathcal{L}} L_{clust}(\theta_d; x), \tag{5.6}$$

where $\mathcal{D}$ denotes the set of all nodes in our hierarchical model, and $\mathcal{L}$ denotes the set of leaf nodes.

Implementation Details: The parameters of the shared low level layers and the layers of the parent are kept frozen until we reach the fine-tuning stage of the overall network. Similarly, when adding a new specialist model, only the parameters of the layers that are specific to newly added models are updated. As the architecture of the specialist model

alternates between the levels in the hierarchy, a specialist is initialized with its grandparent specialist whom it shares the architecture with. We initialized our base models with pre-trained networks on ImageNet, and then fine-tuned for the target dataset, with an exception in the experiment on CIFAR-100, where we trained the base model from the scratch until its accuracy reached the performance for the same model reported in (Yan et al., 2015; Ahmed and Torresani, 2017). The number of confusing clusters K are set to 2 for all levels in the hierarchy. The threshold τ for the routing function is empirically selected as 0.96. We used stochastic gradient descent for the optimization. The deployed learning rate was 0.001. The learning rate is reduced by a factor of 10 when the validation loss plateaus. To combat overfitting, standard data augmentation techniques such as random cropping, scaling, aspect ratio setting (Szegedy et al., 2015), and color jittering (Urban et al., 2016) were applied. We used an image resolution of 224×224. Our proposed network is implemented in PyTorch (Paszke et al., 2017).

5.2 Experiments

We performed quantitative and qualitative evaluation to validate our proposed hierarchy of alternating specialists. We provide quantitative results to demonstrate the benefit of our divide-and-conquer strategy based on feature-based grouping, as well as our proposed alternating architecture. Also, we qualitatively validate our premises on feature-based grouping by visualizing the learned hierarchy. Furthermore, we qualitatively show how region of interest are changed in specialist models as compared to that of the generalist model, which led to correct scene class predictions. For a direct comparison with other tree-structured networks, we also show the results of our architecture on the image classification task of CIFAR-100.

5.2.1 Datasets and evaluation methodology

Dataset: We performed experiments on the widely used SUN database (Xiao et al., 2010). The original number of scene categories in this dataset is 397. However, the majority of classes contain just around 100 example image per category. Such an amount of data is not sufficient for training a deep network, especially considering that the dataset need to be split into disjoint training, validation, and test sets. To alleviate the potential overfitting problem, we create a subset of SUN397 (Xiao et al., 2010), the SUN190 dataset, which consist of classes that contains at least 200 examples, resulting in 48K images in total. Following (Agrawal et al., 2014), we randomly divide the data for training, validation, and test with the proportion of 60%, 10%, and 30%.

We also performed experiments on another publicly available dataset, the Places205 (Zhou et al., 2014b), which contains 2.5M images. This dataset has more examples per class as compared to the SUN190 dataset. However, we use SUN190 dataset for comprehensive study as its mid-size allows us to carry out many different design choices. For the Places205 dataset, we treated the validation set as our test set.

Finally, for the comparison with the existing tree-structured networks, we also trained and evaluated our architecture on the CIFAR-100 dataset, a standard image classification benchmark which contains 60K images in total.

Evaluation Metric: Following the standard protocol of SUN dataset (Agrawal et al., 2014; Zhou et al., 2014b) as well as Places205 dataset (Zhou et al., 2014b), we report one-vs.-all classification accuracy averaged over all classes. We report both top-1 accuracy and top-5 accuracy for these datasets, and top-1 for the CIFAR-100 dataset. In all our experiments, test images for evaluation were resized to a resolution of 224×224. Statistics are collected under single-view testing, i.e., no averaging of multiple crops (Agrawal et al., 2014; Zhou et al., 2016; Krizhevsky et al., 2012) were performed.

Base model: For our base network, we used the AlexNet* architecture (Krizhevsky, 2014), which is a slimmer version of the original AlexNet proposed in (Krizhevsky et al., 2012). We

also propose to let the specialist share the parameters with the lower layers of the generalist up to `conv4`. We use the same architecture for ordered pooling as the base network does. For orderless pooling, we used an architecture similar to AlexNet-GAP-Wide (Bau et al., 2017). We keep the layers of AlexNet* up to `conv5` and add a `conv6` layer with 768 3×3 filters, with a global average pooling layer between `conv6` and `fc7`.

5.2.2 Scene classification results

Comparison with category-group-based network of experts: In order to evaluate our premise that specialists trained on confusing clusters are better than those trained on coarse categories, we compare with a network of experts based on coarse categories. In particular, we compare a two-level hierarchical model similar to HD-CNN (Yan et al., 2015), but with AlexNet* (Krizhevsky, 2014) (HD-CNN*) as a baseline for a fair comparison with our method. For this baseline method, spectral clustering was performed on covariance matrix of class predictions for discovering the set of confusing classes as in (Hinton et al., 2015; Murthy et al., 2016). The final prediction is made using the weighted average of predictions as in (Yan et al., 2015). We experimented with a different number of clusters of 2, 4, and 8 for this model. Furthermore, we compare our approach with a simple ensemble model, where the models are trained with different initializations and the predictions are averaged. We also report the performance of the fine-tuned single AlexNet* (Krizhevsky, 2014) model, which is also our generalist model at the root of the hierarchy.

In Table 5.5, we compare our performance with aforementioned baselines on SUN190 dataset. All of our models outperform the baselines, where our best model with a 3-level hierarchy achieved a classification accuracy of 66.41% for the Top-1 prediction, exceeding the accuracy of the coarse-category-based model (HD-CNN*) by 2.76%. The performance of our proposed model consistently improves as we increase the number of levels in the hierarchy. In contrast, HD-CNN* only has marginal improvements in the Top-1 accuracy, while the Top-5 accuracy drops as the number of clusters increases. This demonstrates the effectiveness of our

Table 5.1: Scene classification accuracy. All compared models are based on AlexNet* (Krizhevsky, 2014) architecture. Statistics are collected under single-view testing.

Dataset	Method		Top-1 acc.	Top-5 acc.
SUN 190	AlexNet* (Krizhevsky, 2014) fine-tuned	Lev 0	63.46	89.18
	Proposed (Alternating architecture)	Lev 1	66.13	89.66
		Lev 2	66.37	89.85
		Lev 3	**66.41**	**89.96**
	Model 1 (Ordered pooling only)	Lev 1	64.02	89.27
		Lev 2	64.33	89.44
		Lev 3	64.43	89.48
	Model 2 (Orderless pooling only)	Lev 0	61.79	88.14
		Lev 1	62.71	88.54
		Lev 2	63.14	88.76
		Lev 3	63.08	88.69
	HD-CNN* (Yan et al., 2015)	K = 2	63.11	88.81
		K = 4	63.62	87.64
		K = 8	63.65	84.08
	Simple Ensembles	N = 2	64.19	89.47
		N = 4	64.66	89.72
		N = 6	64.82	89.85
		N = 8	64.99	89.96
Places 205	AlexNet* (Krizhevsky, 2014) fine-tuned	Lev 0	48.67	79.24
	Proposed	Lev 1	50.21	79.82
		Lev 2	51.42	80.67
		Lev 3	**51.54**	**80.76**

model in discovering the correct hierarchical organization of image data while overcoming the intra-class variation issues inherent in conventional tree-structured models. We also observe that while our model achieves well-balanced clusters, the spectral clustering resulted in high bias in the number of classes per coarse category. The simple ensembles also underperfoms our method, despite the fact that it averages predictions of all models in the ensemble, while our method outputs the prediction of a single specialist model. We also show the scene classification performance on the larger Places205 dataset on the same table (Table 5.5). Our proposed approach has the improvements of 2.85% over the base model at Top-1 accuracy, which shows the generalizability of our approach.

Table 5.2: Scene classification performance of the AlexNet* (Krizhevsky, 2014) with different global pooling schemes on SUN190 dataset.

Global Pooling Strategy	Top-1 acc.	Top-5 acc.
ordered	63.46	89.18
order-less	61.03	87.71
early fusion	62.29	88.11
late fusion	64.45	89.36
Proposed	**66.41**	**89.96**

Table 5.3: Statistics of the AlexNet* (Krizhevsky, 2014) with global ordered/orderless pooling on SUN190. The IoU of the correct predictions suggests that the two are complementary.

IoU of correct predictions	0.781
union of correct predictions	0.729
prediction overlap ratio	0.732

Table 5.4: Comparison with other tree-structured models on CIFAR-100 dataset. All compared models are based on NIN-C100 (Lin et al., 2013) architecture. Statistics are collected under single-view testing.

Method	hierarchy levels	#model choices	#model selected	#models invoked	Accuracy (%)
NIN-C100 (Lin et al., 2013)	0	1	1	1	64.73
Proposed	1	3	1	1–2	67.32
	2	7	1	1–3	67.61
	3	15	1	1–4	**67.70**
HD-CNN (Yan et al., 2015)	1	9	9	10	65.64
NofE (Ahmed et al., 2016)	1	10	1	2	65.91
BranchConnect	1	10	1	10	66.10
(Ahmed and Torresani, 2017)			5	10	66.45

Benefits of Alternating Architecture: The performance of architectures with global ordered pooling and global order-less pooling are shown in Table 5.2. Both models achieve similar accuracy, while global ordered pooling shows slightly better performance. On the other hand, as shown in Table 5.3, the IoU of the correct prediction is 0.781. This quantitatively validates our assumption that the two architectures are complementary. We also show the performance of fused features from these two architecture in Table 5.2, one with early fusion that concatenates two representations before the last fully connected layer, and the one with late fusion where two predictions are averaged. The late fusion achieves better performance than both global ordered pooling and global orderless pooling, however, does not reach the classification accuracy of our proposed alternating architecture.

Furthermore, in Table 5.5, we compare our Hierarchy of Alternating Specialists with another version of our model that has the same exact structure, but using a non-alternating

architecture (i.e., all specialists in the hierarchy have the same exact architecture). In particular, we compare with Model 1, where all models use the architecture with ordered pooling, and Model 2 that uses the architecture with orderless pooling. Both models were trained with the same training protocol as our proposed model. While the performance of our model with alternating architecture improves with an increasing depth of the hierarchy, models with non-alternating architecture has no observable performance gain. We suspect that this is due to the fact that our alternating architecture is better at yielding confusing clusters, by using two different types of feature sets which capture both the coarse spatial information and the transient objects in the scene.

5.2.3 Comparison with other tree-structured models

For a direct comparison with other tree-structured networks, we show the results of our architecture on the image classification task of CIFAR-100. We compare with recent HD-CNN (Yan et al., 2015), NofE (Ahmed et al., 2016), and BranchConnect (Ahmed and Torresani, 2017). All these methods train their experts on coarse categories while our method alone uses *confusing clusters*. The NofE (Ahmed et al., 2016), and BranchConnect (Ahmed and Torresani, 2017) requires additional network or layers to be used for gating. We show the recalls reported in their original paper, except for NofE (Ahmed et al., 2016) in which we used the recalls reported in (Ahmed and Torresani, 2017) in order to match the performance of the base model for a fair comparison. All models are based on NIN-C100 (Lin et al., 2013) architecture. The results are shown in Table 5.5, where we illustrate the number of models to choose from, the number of selected models, the total number of invoked models, as well as the performance of the compared model. Our approach outperforms all baseline methods despite the fact that it outputs the prediction of a single specialist network, rather than averaging predictions of a multiple model. Our method also invokes the least number of models. In particular, our model outperforms the best baseline BranchConnect (Ahmed and Torresani, 2017) with significantly fewer models invoked.

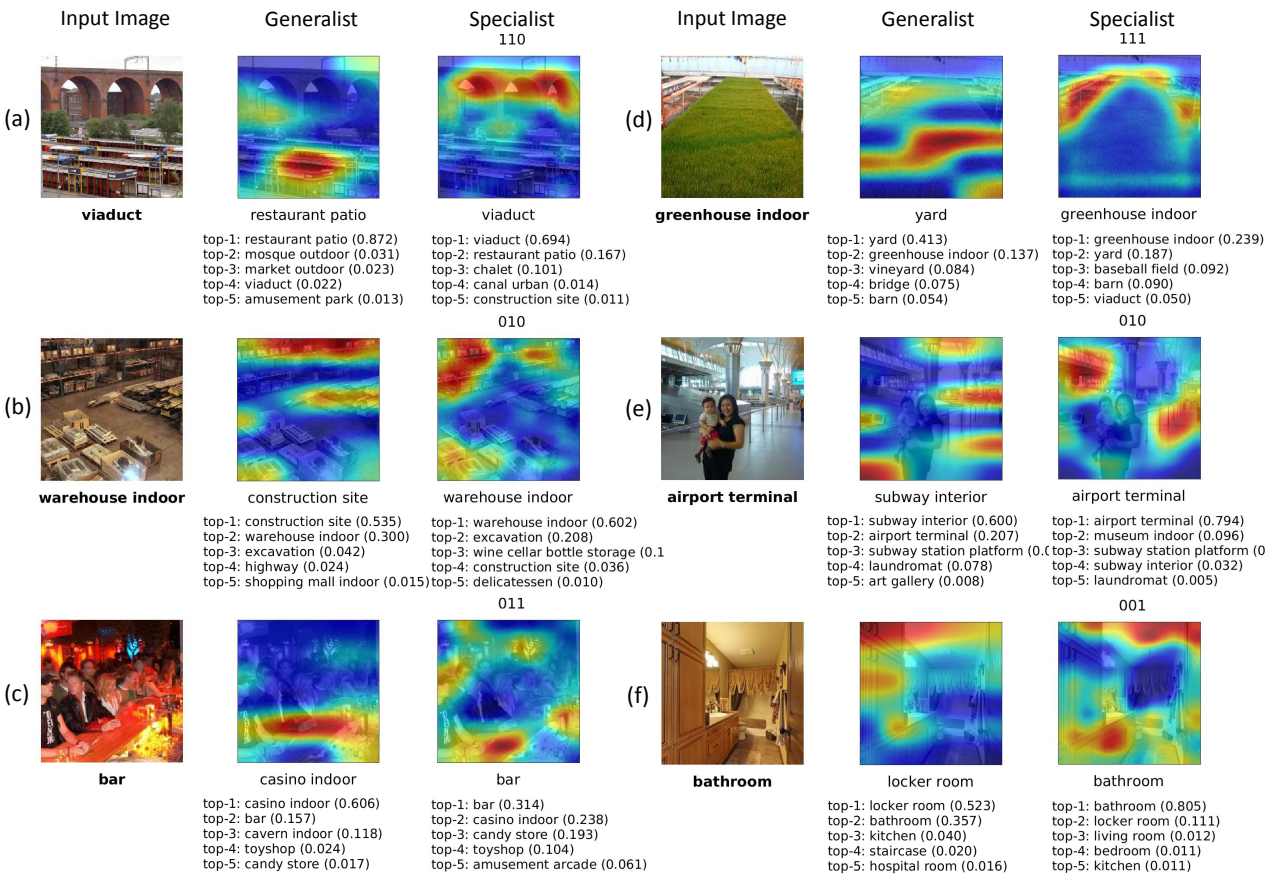


Figure 5.5: (left) Input images and ground-truth category. The top-5 predictions and the visualization of class activation maps (CAM) of the top predicted class for the generalist (center) and the selected specialist (right)

5.2.4 Visualization of Learned Hierarchy of Specialties

We visualize the learned hierarchy of images in Fig. 5.6. For each centroid of the discovered confusing clusters that the specialists were trained on, we depict the top 10 nearest neighboring images in the training set. We observe that each cluster consists of visually coherent and easily confusable images from different scene classes. Instances of the same class appear in different clusters that are visually distinct from each other. For example, a subset of the kitchen class that is similar to bathroom images with base cabinets, which appear in the cluster for the Specialist 001, while the subset of the same category is deemed similar to restaurants and bars, which are found in the cluster corresponding to Specialist 10. This visualization strongly supports our underlying idea of *confusing clusters*.

5.2.5 Comparison of Regions of Interest (ROI)

The benefit of our proposed architecture lies in the specialists' ability to discriminate between classes based on subtle details for images that falls into their specialty. As specialists are trained on the subset of data which reflects their specialty, it evolves to focus on such details to better accommodate the classification task at hand. To illustrate these changes in activation patterns, we investigated how the regions of interest (ROI) of the specialist models differ from those of the generalist models. We visualize the corresponding class activation maps (CAM) (Zhou et al., 2016) for the specialists and the generalists. Since CAMs show the regions that contributed to the prediction of the class in question, we are able to tell which regions in the image contributed to the correct (or the incorrect) prediction. For computing CAM, we used a similar scheme as in (Selvaraju et al., 2017; Zhou et al., 2016):

$$M^c{}_{CAM}(x, y) = \sum_k w^c{}_k(x, y) f_k(x, y) + b^c, \tag{5.7}$$

where $f_k(x, y)$ is the k-th filter activation in the last convolutional layer at the spatial location (x, y). The $w^c{}_k(x, y)$'s are the weights that correspond to class c for unit k, which, along with b^c, is estimated from combining the linear transformations in the fully-connected layers, ReLU. For the orderless pooling, $w^c{}_k$ is the constant for all locations (x, y). $w^c{}_k(x, y)$ and b^c are estimated from combining the linear transformations in the fully-connected layers, which follows the convolutional layers, excluding the rectified linear units (ReLU).

Fig. 5.5 shows the CAMs of the top predicted class for both the generalist and the specialists. We only show examples where the specialists with the depicted results are invoked by our routing function. We observe that the specialists are good at focusing on fine-grain details as compared to the generalist models. For example, in Fig. 5.5 (b), the generalist reasonably predicted the scene category as construction site, based on the construction materials on the right side of the image. However, the specialist was able to focus more on the boxes, predicting the correct scene class of warehouse indoor. In Fig. 5.5 (d), generalist

predicted yard for the scene class, based on the grass field in the center of the image. However, the specialist payed more attention to plants and frames on the sides to predict the correct class of greenhouse indoors.

5.2.6 Ablation Study

5.2.6.1 Benefits of having a deeper hierarchy

To assess the benefit of expanding the hierarchy in our proposed method, rather than having a flat two-level structure (with a single level of specialists) as in the existing state-of-the-arts, we compare the performance of the two in Table 5.5 on the SUN-190 datset. We denote the flat version of our model as HAS-flat, with respect to our proposed Hierarchy of Alternating Specialists (HAS) with multiple hierarchical levels. We observe that our HAS model with a 3-level hierarchy of specialists clearly outperforms the HAS-flat with various numbers of child specialists.

5.2.6.2 Comparison of confusing-cluster-based and coarse-category-based specialists in flat models

To further demonstrate the benefit of training specialists on *confusing clusters*, we performed a controlled experiment where we restricted our models to a flat structure and using a *non-alternating* architecture (HS-flat, Fig. 5.8 (c)). This allows us to directly compare against the previous methods using coarse categories that do not have multi-level hierarchy or the alternating architecture. Table 5.6 shows the result of our comparison on the CIFAR-100. The performance of our models with the alternating architecture (HAS-flat, Fig. 5.8 (b)) are shown in the bracket for reference. Even without both the multi-level hierarchy and the alternating architecture, our specialists trained on confusing clusters (HS-flat) perform better than previous methods that train specialists on coarse categories (Lin et al., 2013; Yan et al., 2015; Ahmed et al., 2016), while our proposed model (HAS, Fig. 5.8 (a)), achieves the best performance.

Table 5.5: Comparison of hierarchical vs. flat structure (HAS vs. HAS-flat, Fig. 5.7 (a,b)). #specialist denote the total number of specialists in the model. We report scene classification accuracy on SUN-190 dataset, using AlexNet* (Krizhevsky, 2014) architecture as a base model.

Method	hierarchy	#specialist	Top-1 acc.	Top-5 acc.
AlexNet* (Krizhevsky, 2014) fine-tuned	lev 0	0	63.46	89.18
HAS-flat	lev 1	3	65.53	89.47
		4	65.48	89.52
		5	65.44	89.46
		8	65.53	89.60
HAS (Proposed)	lev 1	2	66.13	89.66
	lev 2	6	66.37	89.85
	lev 3	14	**66.41**	**89.96**

5.2.7 Visualization of the mini-batch soft k-means on MNIST

To illustrate how well our introduced mini-batch soft k-means (described in Sec. 3.2 of the paper) works, we show its result on the MNIST (LeCun, 1998) dataset using LeNet (LeCun et al., 1998) in Fig. 5.9. We initialize the centroids with randomly chosen data points. We first show the easy case, where all model parameters θ are already optimized under the classification loss and frozen, and only the centroids μ are updated based on the clustering loss (Eqn. (3) of the paper). As shown in Fig. 5.9 (a), the method successfully finds the centroids for each cluster. However, the case we are interested in is when both the parameters θ and the centroids μ are optimized jointly under both the classification loss and the clustering loss, as in our scenario for fine-tuning (Eqn. (6) of the paper). The result for this case is shown in Fig. 5.9 (b), where the mini-batch soft k-means again correctly found the centroids for each cluster. Note that features are L2-normalized in our method (f_θ of the paper). When we use unnormalized features for clustering, the method fails to find the correct centroids as depicted in Fig. 5.9 (c).

Table 5.6: Comparison of *confusing-cluster*-based specialists, HS-flat (non-alternating) and HAS-flat (alternating), and the coarse-category-based specialists ((Lin et al., 2013; Yan et al., 2015; Ahmed et al., 2016)). All these models have a flat two-level structure with a single level of specialists, using NIN-C100 (Lin et al., 2013) architecture as a base model. The performance of our proposed HAS is also shown. We report image classification accuracy on CIFAR-100 (Krizhevsky and Hinton, 2009) using single-view testing.

Method	hierarchy	#specialist	Top-1 accuracy
NIN-C100 (Lin et al., 2013) fine-tuned	lev 0	0	64.73
HD-CNN (Yan et al., 2015)	lev 1	9	65.64
NofE (Ahmed et al., 2016)	lev 1	10	65.91
BranchConnect (Ahmed and Torresani, 2017) (best)	lev 1	10	66.45
HS-flat (HAS-flat)	lev 1	2	66.92 (66.96)
		3	66.81 (66.94)
		4	66.78 (66.89)
		5	66.79 (66.76)
		8	66.64 (66.89)
		10	67.00 (66.97)
HAS (Proposed)	lev 3	14	**67.70**

5.2.8 Computational Time

Our model can be run in parallel or sequentially. Running sequentially minimizes the number of invoked models, thus saving memory at the expense of time. The opposite is true when running in parallel. Let $t_A = t_l + t_u$ be the execution time for the base model, where t_l and t_u denote the time spent on the lower layers (shared in our method) and the upper layers. Let t_r be the time for computing γ (Eqn. (1)) and L the hierarchical levels. When run sequentially, the best case is $t_A + t_r$ when routed to the generalist, while the worst is $t_l + L \cdot (t_u + t_r)$ when routed to a leaf specialist. On an NVIDIA GTX1080Ti with batch size 512 using AlexNet*, it takes 105, 121, and 138ms for our models with $L = 1, 2, 3$, respectively. AlexNet* takes 87ms. When fully parallelized, each model in the hierarchy is run in parallel, then a model is selected, which takes $t_A + t_r + L \cdot t_c$, where t_c is for AND operation on γ. It takes 89ms for all our models ($L = 1, 2, 3$).

5.3 Conclusion

In this chapter, we introduced a novel hierarchy of specialist models for tackling intra-class variation through a divide-and-conquer strategy. The global feature pooling strategy of the specialist model alternates at each level to account for both coarse scene layout and transient objects, which are both essential for accurate scene classification. For defining the area of specialties for each specialist model, we propose to discover confusing image clusters in an unsupervised manner, without any manually-defined subclass information. In particular, we perform clustering based on the learned features of the parent, thereby obtaining image clusters that are visually coherent and confusing at the same time. We experimentally show that our model achieves better performance than those trained on image clusters from a class-based grouping. Through our novel routing function, only a fixed number of models are invoked for both training and testing. We also propose to use soft k-means for end-to-end learning of the hierarchy. Our algorithm is applicable to a variety of CNN models and different vision tasks.

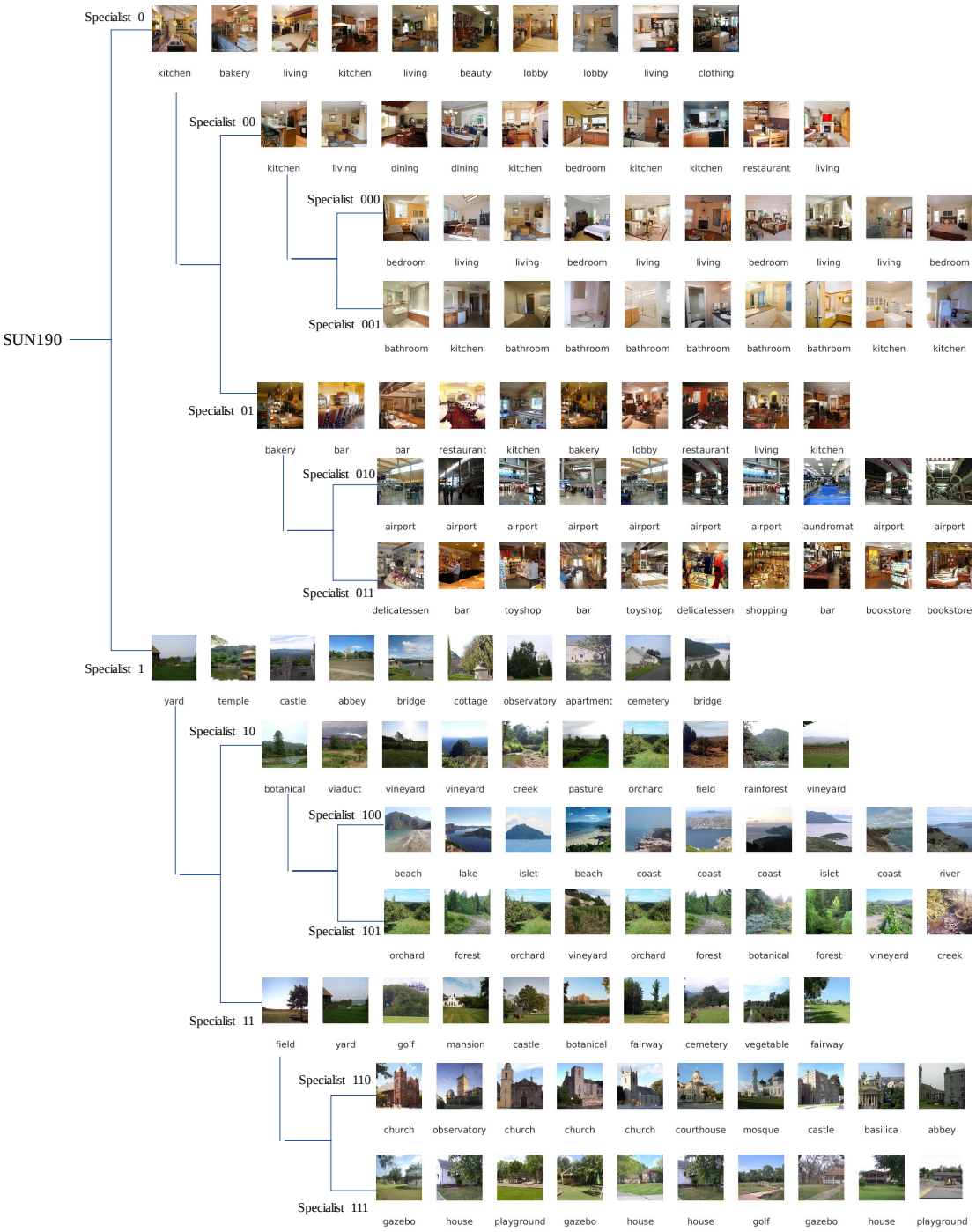


Figure 5.6: Visualization of the learned hierarchy on the SUN190 dataset. A three level hierarchy is shown, with the 10 top images associated with each specialist.

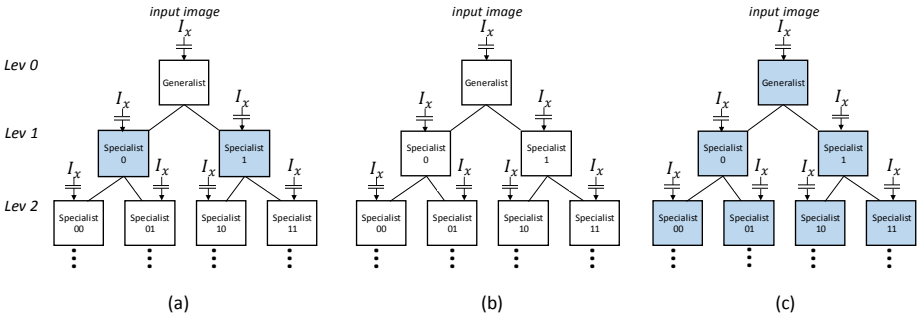


Figure 5.7: (a) Our proposed Hierarchy of Alternating Specialists. (b) Model 1, using global-ordered pooling architecture only. (c) Model 2, using global-orderless pooling architecture only. The white and the blue shaded box denote network architectures with different global pooling strategy.

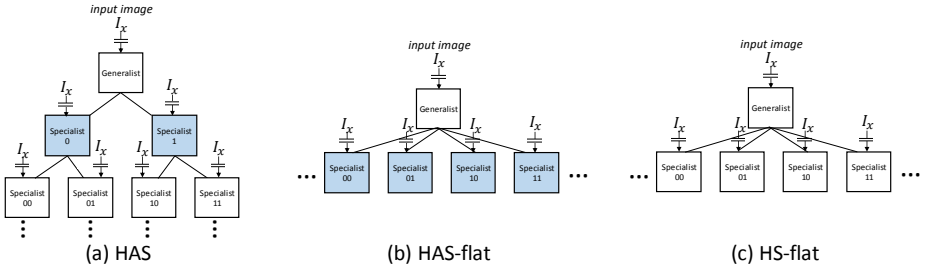
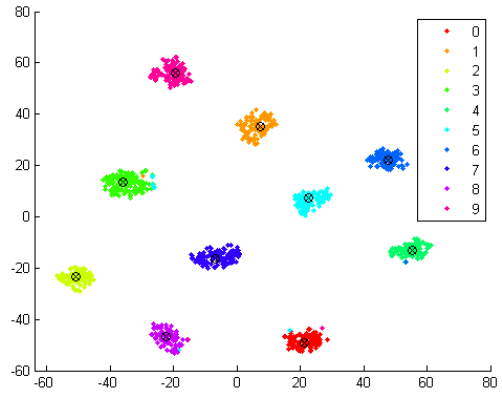
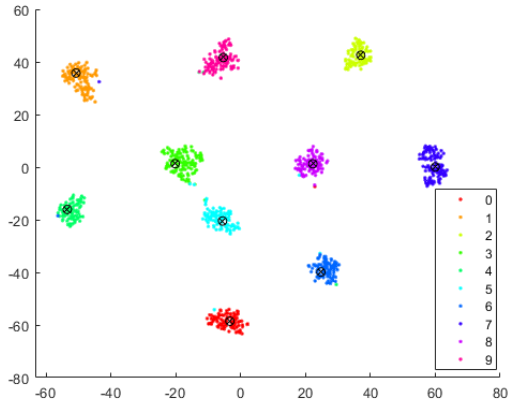


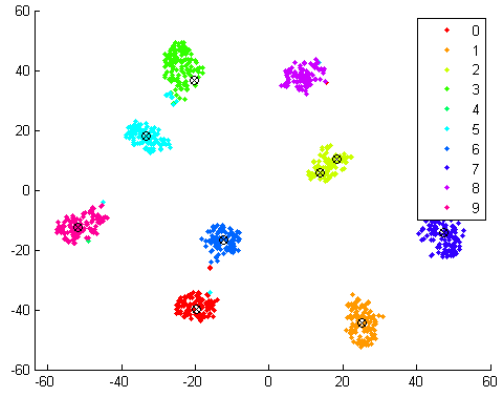
Figure 5.8: (a) Our proposed Hierarchy of Alternating Specialists (HAS). (b) HAS-flat, a flat version of our model with single level of specialists. (c) HS-flat, a flat version of our model without the *alternating* architecture. The white and the blue shaded box denote network architectures with different global pooling strategy.



(a)



(b)



(c)

Figure 5.9: Mini-batch soft k-means result on MNIST (LeCun, 1998) dataset. Each centroids are depicted as $\otimes$. (a) Pre-trained CNN with frozen parameters (only centroids μ are updated) with normalized features. (b) Joint optimization of classification and clustering loss (updating both CNN parameters θ and centroids μ) with normalized features, and with (c) unnormalized features.

CHAPTER 6: CONCLUSION

In this dissertation, we focused on building fully-automatic computer systems capable of performing large-scale image search and scene recognition purely based on visual information. Content-based image retrieval is a challenging problem, because of large variability between the query and the relevant database image, as well as severe visual overlap between irrelevant images. We need image representations that focuses on relevant image content, as opposed to irrelevant and distracting information. Achieving this in a data-driven manner allows exploiting optimal priors beyond limited human supervision, thus leading to a better performance. Hence, our main focus was on understanding underlying factors for discriminating relevant and irrelevant image content and designing image representations that adaptively weigh image content based on those factors. Learning those representations led to unveiling of the hidden factors. In particular, we explored image representations that account for a visual element adaptively based on (1) local image context, (2) semi-global image context, and (3) global image context. We have demonstrated that the our proposed adaptive image representation improves the performance over the state-of-the-arts and yields interesting by-products that gives us insights about our visual world. Our methods are also generalizable to other retrieval and recognition tasks. In the following, we summarize our innovations in each chapter.

In Chapter 3, we introduced the per-bundle vector of locally aggregated descriptors (PBVLAD) as a novel representation for bundled local features that is effective for both learning to predict features and image retrieval. We present a way to predict a data-driven notion of good local features for image geo-localization. We showed that by selecting features based on predictions from learned classifiers, geo-localization performance can be improved.

Additionally, although our approach makes no assumption on features that are useful for geo-localization, we observed semantic relationships emerge through the learning process.

In Chapter 4, we proposed a novel model for learning image representations that integrates context-aware feature reweighting in order to effectively focus on regions that positively contribute to geo-localization. In particular, we introduced a Contextual Reweighting Network (CRN) that predicts the importance of each region in the feature map based on the image context. This model is learned end-to-end for the image geo-localization task, and requires no annotation other than image geo-tags for training. In experimental results, the proposed approach significantly outperformed the previous state-of-the-art on the standard geo-localization benchmark datasets. We also demonstrated that our CRN discovers task-relevant contexts without any additional supervision.

In Chapter 5, we designed a hierarchical generalist-specialist model that automatically builds itself based on the unsupervised discovery of confusing clusters in a coarse to fine manner. The confusing clusters allow specialists to focus on subtle differences between images that are visually similar and confusable to their parents. We also proposed a novel alternating architecture that effectively takes advantage of two complementary representations, which captures spatial layouts and transient objects. Experimental results demonstrate that our method significantly outperforms the baselines including tree-structured models based on coarse categories. Our method is generalizable to other categorization tasks, and is applicable to any CNN architecture.

CHAPTER 7: FUTURE WORK

There are various areas to be explored in the future to extend and to overcome the limitations of methods presented in this thesis.

7.1 Learned Hierarchy of Specialist for Visual Feedback in Interactive Search

Imagine using an image search engine, or browsing an online shopping site. Despite numerous filters on the pane showing different types of attributes, it is often difficult to refine the search result to focus on what you are interested in. The text-based attributes are helpful, but are limited to capture diverse variations of our visual world. While most visual elements rarely form one-to-one relationship with textual descriptions, existing work on automatic attribute discovery mostly focuses on nameable attributes, analyzing the association of visual elements and textual descriptions (Kovashka et al., 2015; Berg et al., 2010; Vittayakorn et al., 2016).

On the other hand, the recommendations based on the past search result, provided in search engines and commercial websites, are also based on the textual meta data. However, obtaining detailed text description is expensive. Online shopping sites and manufacturers typically hire companies or professionals that annotate detailed textual description for an item. An alternative for this approach is performing visual analysis of previously searched items (or hand-picked results by the user) finding common visual elements in the image list, to compare against other images. Such an approach is called relevance feedback. The problem of relevance feedback is that it is time consuming, because a model is retrained every time for a new query.

Instead, we can utilize the nodes in the learned hierarchy (Chapter 5) to offer a user a novel means for modifying the search result without the need for on-the-fly training. The learned feature spaces and cluster centroids in the node can be interpreted as different modes of similarity and attributes, respectively. Users can select the similarity measure, while manipulating the query representation using the associated centroids (attributes). The learned hierarchy can also be used for effective indexing schemes. Such system can utilize visual attributes beyond the nameable ones, and eliminates the need for detailed text annotations. For user interface purposes, we could use slide bars to manipulate the query representation. Also, for symbolizing each feature space, we could use iconic image snippets. In the case when associated text description is available, we could use iconic text description as well.

7.2 Extension to Scene-Category-Aware Place Recognition

One of the challenges in visual place recognition is the scalability. It is not feasible to compute similarity against all images in the database to perform in real time. However, the search space can be greatly reduced if we know the semantic category of the scene. For example, if we know we are looking at a school, a shopfront, or a ranch house, we can limit the search to database images of the same scene category.

On the other hand, semantic information can also be used for tackling the appearance change over time, which is yet another challenge in place recognition. Semantic information is robust to appearance changes. For example, despite the drastic appearance change of a tree due to change in the season, its semantic category does not change. In our work (Chapter 3 and 4), our learned models are aware of such variance and deem transient visual elements (tree, vehicles, pedestrians) to be not useful or irrelevant for the place recognition task, thus avoiding common pitfalls. However, if the visual elements that are typically useful have changed (e.g., a building's exterior has been renovated), our methods are likely to fail, unless there is a significant overlap with its previous appearance.

Hence, our systems of Chapter 4 and Chapter 5 can be combined for scene-category-aware place recognition. Such a system would be also useful for indoor scenes, where the content of the scenes is subject to drastic changes not only due to illumination and viewpoint changes, but also due to frequent interior updates (e.g., furnitures, posters, decorations). Also, such robustness could reduce the cost for maintaining the dataset, eliminating the need for updating the database frequently.

7.3 Extension of Hierarchy of Specialist to World-Scale Place Recognition

A simple extension of our work on city-scale place recognition in Chapter 4 would be extending it to the world-scale problem, using the hierarchical generalist-specialist model in Chapter 5. Existing coarse-to-fine place recognition systems, such as PlaNet, partition the world by geographic regions. In comparison, our hierarchical model will divide the task in a data-driven fashion. A generalist at the root node would solve easy problems such as locating characteristic landmarks, and harder problems would be assigned to the specialist models. As described in Chapter 5, the specialty of a specialist model is discovered through clustering in the learned visual feature space of the generalist model. Thus, we expect specialist models would focus on places that are visually similar, for example, beaches from various coasts, or buildings from different cities. The task of the specialist model is to find the subtle difference between them to distinguish one region from the other. For instance, the type of vegetation and color of the sand and the water would help distinguish between beaches around the world. Of course, these specialty areas may exhibit stronger association to particular regions in the world at deeper levels in the hierarchy.

7.4 Semantic Retrieval of Complicated Scenes

In this thesis, we have considered relatively well-defined retrieval problems that are easy to specify the ground-truth for a given query. In the follow-up work, however, we would

like to explore the problems where the query contains a complex semantic meaning and the ground truth may be only partially true. For example, imagine a query image depicting a man wearing a cap throwing a frisbee at a park with a dog in a cloudy day. First, it is not clear what the user is interested in the image. User interactions, such as relevance feedback or a bounding box would help to determine the object(s) of interest. Existing approaches use *scene-graphs* (Johnson et al., 2015; Xu et al., 2017) to solve complicated queries by first detecting each object separately, then parsing it into a graph the detected objects and their relationships. A graph inferencing technique is used for grounding the graph to the images. As a result, the images with the most corresponding components in the most similar spatial configurations would obtain the highest similarity score. However, we would like to explore an alternative approach for capturing the essence complicated scene. Instead of detecting each object separately and representing the image as a graph, we want to process the image as a whole and discover frequently occurring visual concepts in a data-driven way, which may contain chunks of objects, their coarse spatial configuration, and parts of the background. The visual elements that construct these concepts should be tightly correlated with each other, in order to help minimizing the user interaction, but at the same time, the overall representation would need to be decomposable to allow the users to manually include or exclude certain visual elements.

7.5 Other Directions

We have focused on image retrieval problem where we only have one image query. However, it would be interesting to investigate multi-query image retrieval.

Also, a drawback of our approach is that we need a lot of image data, although we take self-supervised approach for adaptive representation. In many real-world scenarios, the target retrieval application may not have enough images to begin with. Thus, it would be worthwhile to explore directions for domain adaptation.

APPENDIX A: FAST NEAREST NEIGHBOR SEARCH

In this thesis, we have focused on learning an image embedding, where the distance in the embedding space serves as a measure for the visual similarity between the two images. However, it is computationally expensive and often infeasible to compute the distance between the query and all images in the database. A myriad of work has been done to improve the scalability, using tree-search, hashing, and quantization. This section reviews some of the popular fast nearest search techniques, which can be also used with our proposed approaches, unless otherwise specified.

A.0.1 Indexing using Vocabulary Tree

A vocabulary tree (Nister and Stewenius, 2006) used with inverted index provides fast nearest neighbor search for bag-of-visual-words representations (Sivic and Zisserman, 2003). A *vocabulary tree* is a hierarchical quantizer where each leaf of the tree corresponds to a visual word. The *inverted index*, on the other hand, is a data structure that maps visual words to the indices of images that contains each visual word. During the query phase, local features extracted from the query image are assigned to a visual word through the vocabulary tree in $O(KL)$, where K and L are the number of centroids per level and the tree depth, respectively. Also, by incrementing the number of co-occurring word for the list of images returned by the inverted index for each visual word in the query image, the nearest neighbor are found in an efficient manner. However, this method cannot be directly used with our approach. Instead of accounting features that are assigned to a visual words equally, our image representations considers second order statistics (Chapter 3), use soft assignment (Chapter 4), or do not use the visual word approach at all (Chapter 5).

A.0.2 KD-Tree

A KD-Tree (Friedman et al., 1977) takes a vector representation and finds approximate nearest neighbors via tree search. First, a tree is constructed by recursive partitioning of the database into two sets, by spitting along one of the k feature dimensions at the median, until the partitioning criteria—the number of data points, or the maximum distance between the points at the node—is reached. It is a generalization of a binary tree to k-dimensional space. For a given query image, the tree is recursively traversed through depth-first search. When a leaf is reached, it finds nearest neighbors in the leaf bin. If a new nearest neighbor is found, then it updates the nearest neighbor list and the *bounding box* which is created from the distance to the nearest neighbors. A subtree is not traversed if there it has no overlap with the bounding box.

A.0.3 Locality Sensitive Hashing (LSH)

A locality sensitive hashing (Datar et al., 2004) maps high-dimensional vector to a low dimensional binary space using multiple hash functions. Each hash function should satisfy the property that the similarity between two images is equal to the probability they will map to the same bucket: $P[h(d_1) = h(d_2)] = Sim(d_1, d_2)$. An example of such hash function is a random projection, while it can be also learned to produce better results. Examples of the learned functions include using K-means (Paulevé et al., 2010) (unsupervised), and semantic hashing (Salakhutdinov and Hinton, 2009) (supervised). At the query phase, the hash keys for the query image is computed. Then, the database images whose hash keys overlap with the query are returned using inverted index as candidates. Finally, a direct match is performed against all the candidates. It has a sub-linear search time. A Multi-Probe LSH is an extension of LSH, where multiple buckets that are likely to contain the nearest neighbors are probed for each hash function.

A.0.4 PQ Quantization

PQ quantization (Jegou et al., 2011) approximates the distance between two vector representation by using a small lookup table. A d-dimensional vector v is first split into M sub-vectors v^m of $\frac{d}{M}$ dimensions. For database vectors, each sub-vector is then separately quantized using K centroids $\cup_{k=1}^{K} c_k^m$. The idea is to use a lookup table that stores the distance to the k centroids for each sub-vectors of the query vector v_q, that is, $\|{v_q}^m - {c_k}^m\|^2$, in order to approximate the direct distance $\|v_q - c_k\|^2$ as $\sum_m \|{v_q}^m - {c_k}^m\|^2$. The complexity for computing the lookup table is $O(DK)$, which is much cheaper than the direct distance computing of $O(DN)$, where N is the number of database images and $N >> K$.

APPENDIX B: GEOMETRIC TRANSFORMATIONS

For automatic generation of training dataset in Chapter 3 and 4, we count the number of inlier SIFT (Lowe, 1999) matches to determine whether a reference image depicts the same place as the given training query. For this purpose, we use RANSAC (Fischler and Bolles, 1987) which iteratively finds the geometric transformation that most fits the given putative matches. In this section, we review basic 2D planar transforms.

Translation

Translation transform shifts the set of points $P = \{(x, y)\} \in R^2$ uniformly by a given 2D vector $T = (t_x, t_y)$, such that $P' = \{(x + t_x, y + t_y)\} \in R^2$.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.1}$$

Rotation

Rotation transform rotates the set of points $P = \{(x, y)\} \in R^2$ uniformly by a given angle θ with respect to the origin, such that $P' = \{(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)\} \in R^2$.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.2}$$

Scaling

Scaling shrink or enlarge the distance of each points $p = (x, y)$ with respect to the origin uniformly by the given scalar values s_x and s_y, such that $p' = (s_x x, s_y y)$. Isotropic scaling is a special case when s_x and s_y are equal.

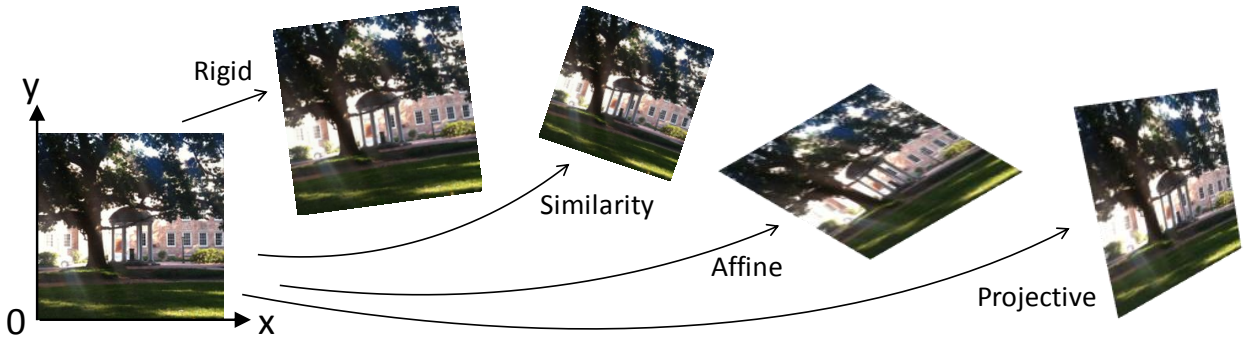


Figure B.1: Geometric transformations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.3}$$

Shear

Shearing shifts the points $p = (x, y)$ along an axis x (y) by the amount proportional to the distance to the other axis y (x), based on scalar value(s) a (and b), such that $p' = (x+ay, y+bx)$.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.4}$$

Rigid Tansform

Rigid transform, also known as 2D Euclidean transform, is a combination of rotation and translation transform. It has 3 degrees of freedom: 1 for the rotation and 2 for the translation.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.5}$$

Similarity Transform

Similarity transform is composed of scaling, rotation, and translation. It has 4 degrees of freedom: 1 for isotropic scaling, 1 for rotation, and 2 for translation. Unlike the rigid transform, it does not preserve the distance between the points.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cdot \cos\theta & -s \cdot \sin\theta & t_x \\ s \cdot \sin\theta & s \cdot \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.6}$$

Affine Transform

Affine transform is composed of rotation, translation, and a deformation. The deformation involves non-isotropic scaling in the direction of an angle ϕ, such that $A = \left[\begin{smallmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{smallmatrix}\right] = R(\theta)R(-\phi)DR(\phi)$, where $D = \left[\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right]$. It has 6 degrees of freedom: 2 for non-uniform scaling, 2 for two rotations, and 2 for translation. Unlike the similarity transform, it does not preserve the angles between the points.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.7}$$

Projective Transform (Homography)

Projective transform is a "general non-singular linear transformation of homogeneous coordinates" (Hartley and Zisserman, 2003). It has 6 degrees of freedom: 2 for non-uniform scaling, 2 for two rotations, 2 for translation, and 2 for *line at infinity*. Unlike the affine transform, parallel lines are not preserved to be parallel under projective transform.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.8}$$

REFERENCES

Agrawal, P., Girshick, R., and Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*.

Ahmed, K., Baig, M. H., and Torresani, L. (2016). Network of experts for large-scale image categorization. In *ECCV*.

Ahmed, K. and Torresani, L. (2017). Branchconnect: Large-scale visual recognition with learned branch connections. *arXiv preprint arXiv:1704.06010*.

Aljundi, R., Chakravarty, P., and Tuytelaars, T. (2017). Expert gate: Lifelong learning with a network of experts. *CVPR*.

Almahairi, A., Ballas, N., Cooijmans, T., Zheng, Y., Larochelle, H., and Courville, A. C. (2015). Dynamic capacity networks. In *CoRR*.

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*.

Arandjelović, R. and Zisserman, A. (2013). All about vlad. In *CVPR*.

Arandjelović, R. and Zisserman, A. (2014a). DisLocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*.

Arandjelović, R. and Zisserman, A. (2014b). Visual vocabulary with a semantic twist. In *ACCV*.

Aubry, M., Russell, B. C., and Sivic, J. (2014). Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics*, 33(2):14.

Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., and Carlsson, S. (2015). From generic to specific deep representations for visual recognition. In *CVPRW*.

Baatz, G., Saurer, O., Köser, K., and Pollefeys, M. (2012). Large scale visual geo-localization of images in mountainous terrain. In *ECCV*.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations.

Bell, S., Zitnick, C. L., Bala, K., and Girshick, R. (2016). Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *CVPR*.

Berg, T. L., Berg, A. C., and Shih, J. (2010). Automatic attribute discovery and characterization from noisy web data. In *ECCV*.

Bishop, C. M. and Svenskn, M. (2002). Bayesian hierarchical mixtures of experts. In *Uncertainty in Artificial Intelligence*.

Cakebread, C. (2017). People will take 1.2 trillion digital photos this year thanks to smartphones. *Business Insider*.

Cao, S. and Snavely, N. (2013). Graph-based discriminative learning for location recognition. In *CVPR*.

Chen, D., Baatz, G., Koser, K., Tsai, S., Vedantham, R., Pylvanainen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., et al. (2011a). City-scale landmark identification on mobile devices. In *CVPR*.

Chen, D., Tsai, S., Hsu, C.-H., Singh, J. P., and Girod, B. (2011b). Mobile augmented reality for books on a shelf. In *ICME*.

Chen, W., Chen, X., Zhang, J., and Huang, K. (2017a). Beyond triplet loss: a deep quadruplet network for person re-identification.

Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., Reid, I., and Milford, M. (2017b). Deep learning features at scale for visual place recognition. In *ICRA*.

Cheng, G., Zhou, P., and Han, J. (2016). RIFD-CNN: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In *CVPR*.

Cheng, X., Lu, J., Feng, J., Yuan, B., and Zhou, J. (2018). Scene recognition with objectness. *Pattern Recognition*.

Chum, O. and Matas, J. (2010). Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR*.

Chum, O., Perdoch, M., and Matas, J. (2009). Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*.

Crandall, D., Owens, A., Snavely, N., and Huttenlocher, D. (2011). Discrete-continuous optimization for large-scale structure from motion. In *CVPR*.

Cummins, M. and Newman, P. (2008). Fab-map: Probabilistic localization and mapping in the space of appearance. *IJRR*, 27(6):647–665.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *CG*.

Delhumeau, J., Gosselin, P., Jégou, H., and Pérez, P. (2013). Revisiting the vlad image representation. In *ACMMM*.

Deng, J., Berg, A. C., and Fei-Fei, L. (2011). Hierarchical semantic indexing for large scale image retrieval. In *CVPR*.

Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., and Adam, H. (2014). Large-scale object classification using label relation graphs. In *ECCV*.

Dixit, M., Chen, S., Gao, D., Rasiwasia, N., and Vasconcelos, N. (2015). Scene classification with semantic fisher vectors. In *CVPR*.

Dixit, M. D. and Vasconcelos, N. (2016). Object based scene representations using fisher scores of local subspace projections. In *NIPS*.

Doersch, C., Gupta, A., and Efros, A. (2013). Mid-level visual element discovery as discriminative mode seeking. In *NIPS*.

Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. (2012). What makes Paris look like Paris? In *ACMTOG*.

Eggert, C., Romberg, S., and Lienhart, R. (2014). Improving vlad: Hierarchical coding and a refined local coordinate system. In *ICIP*.

Fischler, M. A. and Bolles, R. C. (1987). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.

Frahm, J., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn, E., Clipp, B., Lazebnik, S., et al. (2010). Building Rome on a cloudless day. In *ECCV*.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM TOMS*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *TPAMI*, 38(1):142–158.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Aistats*.

Gong, Y. and Lazebnik, S. (2011). Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*.

Gong, Y., Wang, L., Guo, R., and Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*.

Goo, W., Kim, J., Kim, G., and Hwang, S. J. (2016). Taxonomy-regularized semantic deep convolutional neural networks. In *ECCV*.

Gordo, A., Almazan, J., Revaud, J., and Larlus, D. (2016). End-to-end learning of deep visual representations for image retrieval. *ECCV*.

Gronat, P., Obozinski, G., Sivic, J., and Pajdla, T. (2013). Learning and calibrating per-location classifiers for visual place recognition. In *CVPR*.

Gross, S., Ranzato, M., and Szlam, A. (2017). Hard mixtures of experts for large scale weakly supervised vision. *CVPR*.

Guo, S., Huang, W., Wang, L., and Qiao, Y. (2017). Locally supervised deep hybrid model for scene recognition. *TIP*.

Guo, Y., Zhao, G., Pietikäinen, M., and Xu, Z. (2010). Descriptor learning based on fisher separation criterion for texture classification. In *ACCV*.

Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., and Wu, F. (2012). 3d visual phrases for landmark recognition. In *CVPR*.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Hartmann, W., Havlena, M., and Schindler, K. (2014). Predicting matchability. In *CVPR*.

Hays, J. and Efros, A. (2008). Im2gps: estimating geographic information from a single image. In *CVPR*.

Hays, J. and Efros, A. (2015). Large-scale image geolocalization. In *Multimodal Location Estimation of Videos and Images*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

Heinly, J., Schonberger, J. L., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *CVPR*.

Herranz, L., Jiang, S., and Li, X. (2016). Scene recognition with cnns: objects, scales and dataset bias. In *CVPR*.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *Arxiv preprint arXiv:1503.02531*.

Hiramatsu, Y., Hotta, K., Imanishi, A., Matsuda, M., Terai, K., Liu, D., Zhang, D., Song, Y., Zhang, C., Huang, H., et al. (2018). Cell image segmentation by integrating multiple cnns. In *CVPR*.

Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *CVPR*.

Hwang, S. J. and Sigal, L. (2014). A unified semantic embedding: Relating taxonomies and attributes. In *NIPS*.

Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *CVPR*.

Jain, M., Jégou, H., and Gros, P. (2011). Asymmetric hamming embedding: taking the best of our bits for large scale image search. In *ACMMM*.

Jégou, H., Douze, M., and Schmid, C. (2009). On the burstiness of visual elements. In *CVPR*.

Jégou, H., Douze, M., and Schmid, C. (2010). Improving bag-of-features for large scale image search. *IJCV*.

Jegou, H., Douze, M., and Schmid, C. (2011). Product quantization for nearest neighbor search. *TPAMI*, 33(1):117–128.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., and Schmid, C. (2012). Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *CoRR*.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., and Fei-Fei, L. (2015). Image retrieval using scene graphs. In *CVPR*.

Johnson, T., Fite-Georgel, P., Raguram, R., and Frahm, J.-M. (2010). Fast organization of large photo collections using cuda. In *ECCV*.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the em algorithm. *Neural computation*.

Ju, C., Bibaut, A., and van der Laan, M. J. (2017). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *arXiv preprint arXiv:1704.01664*.

Juneja, M., Vedaldi, A., Jawahar, C., and Zisserman, A. (2013). Blocks that shout: Distinctive parts for scene classification. In *CVPR*.

Kalantidis, Y., Mellina, C., and Osindero, S. (2016). Cross-dimensional weighting for aggregated deep convolutional featuresoo. In *ECCVW*.

Kantorov, V., Oquab, M., Cho, M., and Laptev, I. (2016). Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*.

Kendall, A., Grimes, M., and Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*.

Khan, S. H., Hayat, M., and Porikli, F. (2017). Scene categorization with spectral features. In *CVPR*.

Kiapour, M. H., Han, X., Lazebnik, S., Berg, A. C., and Berg, T. L. (2015). Where to buy it: Matching street clothing photos to online shops.

Kim, H. J., Dunn, E., and Frahm, J.-M. (2015). Predicting good features for image geo-localization using per-bundle vlad. In *ICCV*.

Kim, H. J., Dunn, E., and Frahm, J.-M. (2017a). Learned contextual feature reweighting for image geo-localization. In *CVPR*.

Kim, H. J. and Frahm, J.-M. (2018). Hierarchy of alternating specialists for scene recognition. In *ECCV*.

Kim, J., Park, Y., Kim, G., and Hwang, S. J. (2017b). Splitnet: Learning to semantically split deep networks for parameter reduction and model parallelization. In *ICML*.

Knopp, J., Sivic, J., and Pajdla, T. (2010). Avoiding confusing features in place recognition. In *ECCV*.

Koutaki, G. and Uchimura, K. (2014). Scale-space processing using polynomial representations. In *CVPR*.

Kovashka, A., Parikh, D., and Grauman, K. (2015). Whittlesearch: Interactive image search with relative attribute feedback. *IJCV*.

Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A. and Hinton, G. E. (2011). Using very deep autoencoders for content-based image retrieval. In *ESANN*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*.

LeCun, Y. (1998). The mnist database of handwritten digits.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Li, F., Neverova, N., Wolf, C., and Taylor, G. (2017). Modout: Learning multi-modal architectures by stochastic regularization. In *FG*.

Li, Y., Snavely, N., and Huttenlocher, D. (2010). Location recognition using prioritized feature matching. In *ECCV*.

Li, Y., Snavely, N., Huttenlocher, D., and Fua, P. (2012). Worldwide pose estimation using 3d point clouds. In *ECCV*.

Lim, H., Sinha, S. N., Cohen, M. F., and Uyttendaele, M. (2012). Real-time image-based 6-dof localization in large-scale environments. In *CVPR*.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

Lin, T.-Y., Cui, Y., Belongie, S., and Hays, J. (2015). Learning deep representations for ground-to-aerial geolocalization. In *CVPR*.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*.

Linegar, C., Churchill, W., and Newman, P. (2016). Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *ICRA*.

Liu, Z., Luo, P., Qiu, S., Wang, X., and Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*.

Luo, W., Li, Y., Urtasun, R., and Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. In *NIPS*.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*.

Middelberg, S., Sattler, T., Untzelmann, O., and Kobbelt, L. (2014). Scalable 6-dof localization on mobile devices. In *ECCV*.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *NIPS*.

Mousavian, A. and Kosecka, J. (2015). Deep convolutional features for image based retrieval and scene categorization. *arXiv preprint arXiv:1509.06033*.

Mousavian, A. and Košecka, J. (2015). Semantically aware bag-of-words for localization.

Murdock, C., Li, Z., Zhou, H., and Duerig, T. (2016). Blockout: Dynamic model selection for hierarchical deep networks. In *CVPR*.

Murthy, V. N., Singh, V., Chen, T., Manmatha, R., and Comaniciu, D. (2016). Deep decision network for multi-class image classification. In *CVPR*.

Naseer, T., Oliveira, G. L., Brox, T., and Burgard, W. (2017). Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*.

Naseer, T., Spinello, L., Burgard, W., and Stachniss, C. (2014). Robust visual robot localization across seasons using network flows. In *AAAI*.

Nicas, J. (2017). Youtube tops 1 billion hours of video a day on pace to eclipse tv. *The Wall Street Journal*.

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *CVPR*.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope.

Oliva, A. and Torralba, A. (2007). The role of context in object recognition. *Trends in cognitive sciences 11*.

Paszke, A., Gross, S., Chintala, S., and Chanan, G. (2017). Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. `http://pytorch.org/`.

Paulevé, L., Jégou, H., and Amsaleg, L. (2010). Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*.

Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *ECCV*.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.

Qian, Q., Jin, R., Zhu, S., and Lin, Y. (2015). Fine-grained visual categorization via multi-stage metric learning. In *CVPR*.

Qin, D., Wengert, C., and Van Gool, L. (2013). Query adaptive similarity for large scale object retrieval. In *CVPR*.

Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *CVPR*.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in context. In *ICCV*.

Radenović, F., Tolias, G., and Chum, O. (2016). Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. *ECCV*.

Raguram, R., Wu, C., Frahm, J.-M., and Lazebnik, S. (2011). Modeling and recognition of landmark image collections using iconic scene graphs. *IJCV*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*.

Robertson, D. P. and Cipolla, R. (2004). An image-based system for urban navigation. In *BMVC*.

Sablayrolles, A., Douze, M., Usunier, N., and Jégou, H. (2017). How should we evaluate supervised hashing? In *ICASSP*.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *IJAR*.

Sattler, T., Havlena, M., Radenovic, F., Schindler, K., and Pollefeys, M. (2015). Hyperpoints and fine vocabularies for large-scale location recognition. In *ICCV*.

Sattler, T., Havlena, M., Schindler, K., and Pollefeys, M. (2016). Large-scale location recognition and the geometric burstiness problem. In *CVPR*.

Sattler, T., Leibe, B., and Kobbelt, L. (2011). Fast image-based localization using direct 2d-to-3d matching. In *ICCV*.

Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., and Pajdla, T. (2017). Are large-scale 3d models really necessary for accurate visual localization? In *CVPR*.

Sattler, T., Weyand, T., Leibe, B., and Kobbelt, L. (2012). Image retrieval for image-based localization revisited. In *BMVC*.

Saurer, O., Baatz, G., Köser, K., Pollefeys, M., et al. (2016). Image based geo-localization in the alps. *IJCV*.

Schindler, G., Brown, M., and Szeliski, R. (2007). City-scale location recognition. In *CVPR*.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. (2017). Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *ICCV*.

Shrivastava, A., Malisiewicz, T., Gupta, A., and Efros, A. (2011). Data-driven visual similarity for cross-domain image matching. In *ACMTOG*.

Sibbing, D., Sattler, T., Leibe, B., and Kobbelt, L. (2013). Sift-realistic rendering. In *3DV*.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Singh, S., Gupta, A., and Efros, A. (2012). Unsupervised discovery of mid-level discriminative patches. In *ECCV*.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV*.

Sollich, P. and Krogh, A. (1996). Learning with ensembles: How overfitting can be useful. In *NIPS*.

Somanath, G. and Kambhamettu, C. (2010). Abstraction and generalization of 3d structure for recognition in large intra-class variation. In *ACCV*.

Srivastava, N. and Salakhutdinov, R. R. (2013). Discriminative transfer learning with tree-based priors. In *NIPS*.

Sünderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., and Milford, M. (2015). Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *RSS*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR*.

Taneja, A., Ballan, L., and Pollefeys, M. (2014). Never get lost again: Vision based navigation using streetview images. In *ACCV*.

Tolias, G., Avrithis, Y., and Jégou, H. (2016a). Image search with selective match kernels: aggregation across single and multiple images. *IJCV*.

Tolias, G. and Jégou, H. (2014). Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern Recognition*, 47(10):3466–3476.

Tolias, G., Sicre, R., and Jégou, H. (2016b). Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*.

Torii, A., Arandjelović, R., Sivic, J., Okutomi, M., and Pajdla, P. (2015a). 24/7 place recognition by view synthesis. In *CVPR*.

Torii, A., Sivic, J., Okutomi, M., and Pajdla, T. (2015b). Visual place recognition with repetitive structures. 37(11):2346–2359.

Tu, Z. (2005). Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *CVPR*.

Turcot, P. and Lowe, D. G. (2009). Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshops*.

Tuzel, O., Marks, T. K., and Tambe, S. (2016). Robust face alignment using a mixture of invariant experts. In *ECCV*.

Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., and Richardson, M. (2016). Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*.

Vaca-Castano, G., Zamir, A. R., and Shah, M. (2012). City scale geo-spatial trajectory estimation of a moving camera. In *CVPR*.

Vittayakorn, S., Umeda, T., Murasaki, K., Sudo, K., Okatani, T., and Yamaguchi, K. (2016). Automatic attribute discovery with neural activations. In *ECCV*.

Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *CVPR*.

Wang, L., Guo, S., Huang, W., Xiong, Y., and Qiao, Y. (2017a). Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns. *TIP*.

Wang, X. and Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *ICCV*.

Wang, Z., Wang, L., Wang, Y., Zhang, B., and Qiao, Y. (2017b). Weakly supervised patchnets: Describing and aggregating local patches for scene recognition. *TIP*.

Warde-Farley, D., Rabinovich, A., and Anguelov, D. (2014). Self-informed neural network structure learning. *arXiv preprint arXiv:1412.6563*.

Weyand, T., Kostrikov, I., and Philbin, J. (2016). Planet-photo geolocation with convolutional neural networks. *ECCV*.

Wu, R., Wang, B., Wang, W., and Yu, Y. (2015). Harvesting discriminative meta objects with deep cnn features for scene classification. In *ICCV*.

Wu, Z., Ke, Q., Isard, M., and Sun, J. (2009). Bundling features for large scale partial-duplicate web image search. In *CVPR*.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*.

Xiao, Y., Wu, J., and Yuan, J. (2014). mcentrist: A multi-channel feature generation mechanism for scene categorization. *TIP*.

Xie, L., Hong, R., Zhang, B., and Tian, Q. (2015). Image classification and retrieval are one. In *ICMR*.

Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. (2017). Scene graph generation by iterative message passing. In *CVPR*.

Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W., and Yu, Y. (2015). HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*.

Yoo, D., Park, S., Lee, J.-Y., Paek, A., and Kweon, I. S. (2015). Attentionnet: Aggregating weak directions for accurate object detection. In *ICCV*.

Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. *ICLR*.

Yue-Hei Ng, J., Yang, F., and Davis, L. S. (2015). Exploiting local features from deep networks for image retrieval. In *CVPR*.

Zamir, A. R., Ardeshir, S., and Shah, M. (2014). Gps-tag refinement using random walks with an adaptive damping factor. In *CVPR*.

Zamir, A. R. and Shah, M. (2010). Accurate image localization based on google maps street view. In *ECCV*.

Zamir, A. R. and Shah, M. (2014). Image geo-localization based on multiplenearest neighbor feature matching using generalized graphs. *TPAMI*, 36(8):1546–1558.

Zhang, C., Gao, J., Wang, O., Georgel, P., Yang, R., Davis, J., Frahm, J., and Pollefeys, M. (2014). Personal photograph enhancement using internet photo collections. *TVCG*, (2):262–275.

Zhang, W. and Kosecka, J. (2006). Image based localization in urban environments. In *3DPVT*.

Zhao, B., Li, F., and Xing, E. P. (2011). Large-scale category structure aware image categorization. In *NIPS*.

Zheng, W.-S., Gong, S., and Xiang, T. (2011). Person re-identification by probabilistic relative distance comparison. In *CVPR*.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2014a). Object detectors emerge in deep scene cnns. *ICLR*.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *CVPR*.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2018). Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014b). Learning deep features for scene recognition using places database. In *NIPS*.

Zhou, Z.-H., Wu, J., and Tang, W. (2002). Ensembling neural networks: many could be better than all. *Artificial Intelligence*.

Zhu, C.-Z., Jégou, H., and Satoh, S. (2013). Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV*.