Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Towards Efficient Visual Place Recognition Methods in Challenging Environments by Adaptive Representations

Reem Aljuaidi

This thesis is submitted to the University of Dublin, Trinity College
in fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Supervisor: Dr. Michael Manzke

October 2023

Declaration

I declare that this thesis has not previously been submitted as an exercise for a degree at this, or any other University, and it is my own work.

Reem Aljuaidi

2 October 2023

Acknowledgments

At the end of this pleasant journey, I would like to thank my supervisor, Dr.Michael Manzke, for the patience, guidance, encouragement, and advice that he has provided to me throughout my time as a student and in writing this thesis.

On a personal level, my gratitude goes first to my wonderful parents; without their love, support, and encouragement, I would barely finish this journey. In addition, I would like to thank Ali, without whom it would be impossible for me to attain my academic dreams. I also thank my amazing family: Dina, Hamad, Rayda and Nasser for their patience during the past years and for their moral support.

Finally, I would like to offer my sincere gratitude to the wonderful people and friends I met in the Graphics, Vision, and Visualization (GV2) group.

Abstract

Visual Place Recognition (VPR) is the ability to recognize a place by providing a query image of an unknown location. The goal is to identify an image from a geotagged database of street-side imagery that depicts the same location as the query. In outdoor environments, recognizing a place is challenging due to the visual differences between query and database images. To develop a robust VPR method capable of handling environmental changes, the image representation must possess high discrimination to distinguish relevant from non-relevant features. However, the vast number of features between the query image and the dataset image complicates the computational process. The challenge here lies in finding an efficient way to represent images. The objective of this thesis is to present VPR methods that are resilient to dynamic environmental changes while also being efficient in terms of reducing computational demands. To achieve this goal, this dissertation explores how to create image representations that adaptively focus on specific image content. To this end, four contributions are proposed. The first and second contributions concentrate on developing efficient representation methods for accurate visual place retrieval and recognition systems. We propose methods for reducing the computational cost of calculating similarity between two vectors. As our third contribution, we suggest a hybrid feature that remains robust in the face of environmental changes. Subsequently, we extract valuable features from these hybrid representations to create an efficient VPR system. As our fourth contribution, instead of compelling the algorithm to learn relevant and irrelevant image examples, we propose a method that can predict unique features by learning both relevant and non-relevant features in a data-driven manner. In conclusion, the numerous experiments and analyses conducted in this thesis yield quantitative and qualitative results that are on par with the most advanced VPR and retrieval techniques.

Contents

List of Figures

List of Tables

Abbreviations

ANN Approximate Nearest Neighbor

BoW Bag of Words

CBIR Content Based Image Retrieval

CNN Convolutional Neural Network

CSLBP Complate Center-symmetric Local Binary Patterns

FAST Features from Accelerated Segment Test

GMM Gaussian Mixture Model

GPS Global Positioning System

HOG Histograms of Oriented Gradients

HTC Hull Census Transform

HVLAD Hierarchical Vector of Locally Aggregated Descriptors

IR Image Retrieval

LPR Linguistic Place Recognition

MBVLAD Mini Batch Vector of Locally Aggregated Descriptors

MSER Maximally Stable Extremal Regions

ORB Oriented FAST and Rotated BRIEF

PBVLAD Per Bundle Vector of Locally Aggregated Descriptors

PCA Principal Component Analysis

PCAN Point Contextual Attention Network

SIFT Scale-invariant Feature Transforms

SURF Speeded Up Robust Features

SVM Support Vector Machine

TSHQ Tree Structured Hierarchical Quantization

VLAD Vector of Locally Aggregated Descriptors

VPR Visual Place Recognition

Chapter 1

Introduction

This chapter serves as an introduction to the thesis. The motivation for investigating visual place recognition is discussed first. This is followed by a problem description to engage the readers in the challenges of the problem solution. Following this, the research question is expressed explicitly. Then, the thesis structure is supplied, as well as a quick review of the contents of each chapter. Finally, there is a list of publications and contributions.

1.1 Context and Motivation

Imagine being a tourist in a new place and getting lost at night. You are strolling about and trying to figure out where you are when you come to a stop in front of a shop you visited the day before. If you can identify that store, you will have a good idea of where you are and how to get back to your lodging. By comparing what you are seeing now to a memory of a specific place, your brain has effectively helped you pinpoint your physical location in the world. In the same way, VPR is the process of recalling a previously viewed location solely from visual cues. To put it another way, VPR does a before-and-after comparison of images. For optimal performance, a generic VPR system needs both a visual memory (or map) and the ability to generate localization hypotheses based on recent observations. The ability to efficiently and accurately recall previously viewed locations based on visual input alone has attracted a lot of interest in the fields of computer vision and robotics because of its numerous important applications. These include long-term robot navigation and autonomy [64], image search based on visual content [124], location-refinement given human–machine interfaces [106], and asset-management using aerial imagery [95].

The task of visual place recognition has been presented as an image retrieval (IR) problem, in which images of the same location described in a query image are retrieved from a geo-tagged image database. Figure 1.1 shows the image retrieval steps to solve the place recognition problem. However, in outdoor environments, appearance variability happens at radically diverse time scales, posing a significant barrier for life-long VPR. Natural changes, such as time of day, weather fluctuations, changing seasons, and vegetation growth all contribute to cyclical appearance variations. In addition, human activities create more unexpected changes, such as construction activity, rapid changes in traffic flow, and changing signs, facades, and billboards. This requires a long-term VPR capable of reliably matching two images depicting the same location but with different appearances. Figure 1.2 shows examples of life-long VPR.

In real-time applications such as self-navigation, the VPR system should be efficient and robust against environmental changes. To this end, many approaches address this problem by improving image representation methods [63, 8, 55]. The representation needs to be both discriminative and efficient with regards to resources in order to create an efficient visual place recognition system (e.g., fast search time). However, most current VPR-related efforts do not provide a useful localization procedure. Very few papers create VPR-based techniques for real-time applications, and even fewer do so in environments with significant difficulties [94].

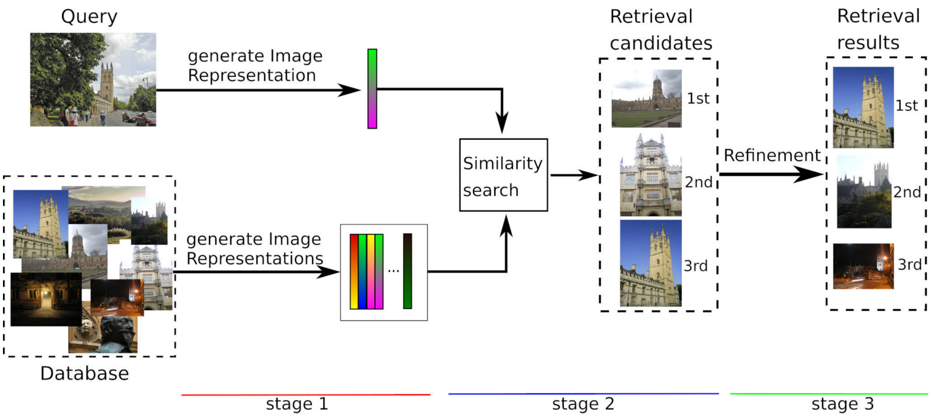


Figure 1.1: Image retrieval is a typical formulation for visual place recognition. The known locations are stored in a database, and a new image to be localized is referred to as a query. The location retrieval process is divided into three logical stages. Image from [150].

The existence of irrelevant image segments or unrelated background elements typically limits the accuracy of VPR systems. Thus, selecting informative features has been proposed for improving VPR

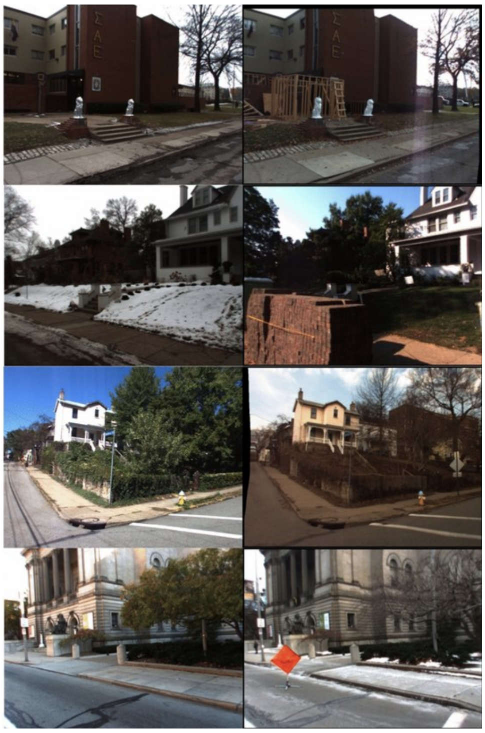


Figure 1.2: An example of the same place images but different appearance during long-life. The image is from [34].

accuracy in challenging environments. However, the majority of the proposed solutions require location-specific learning or database expansion, which are either inefficient or expensive. Moreover, time-consuming computations, such as querying each database image to find misleading or informative features, are insufficiently scalable and, hence, less practical. As a result, these approaches fail to adequately address the challenges presented by the changing nature of databases due to objects such as people flow, growing trees, vehicles, and billboards. Figure 1.3 shows dynamic objects in street-level images as an example. Therefore, the problem of how to recognize similar places in long-term localization must be tackled differently. The work undertaken in this thesis aims to contribute efficient visual place recognition methods for long-term localization.

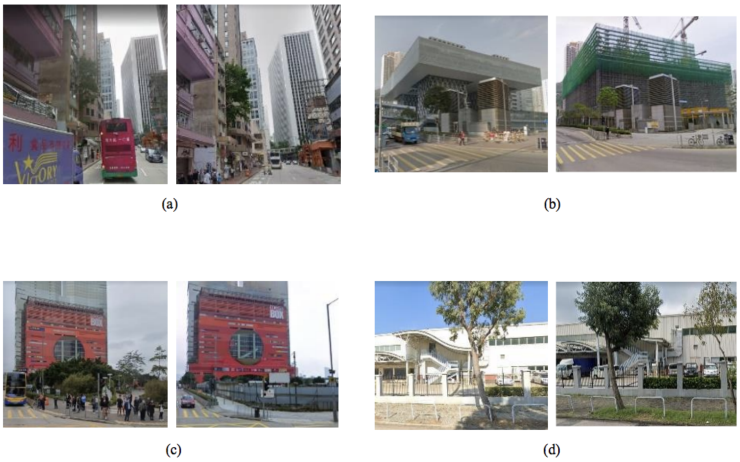


Figure 1.3: Examples of existing issues with street-level image-based VPR. (a) depicts the obstruction of traffic flow. (b) Background occlusions caused by people and trees. (c) New urban building development and renovation. (d) The effect of photo shooting direction [152].

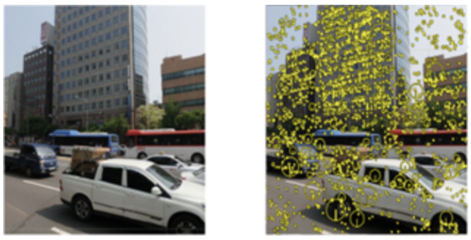


Figure 1.4: An example of features are extraction from street view images [98].

1.2 Research Problems and Scope

The contributions in this thesis can be broadly categorised under the following topics.

1.2.1 Image Representation for Visual Place Retrieval

In this thesis, we propose efficient visual retrieval techniques. VPR should be discriminative and fast for real-time applications. Mobile robots, for example, must traverse unknown environments from given start coordinates to supplied goal coordinates. In such a situation, robots should be ready to quickly change their plans when their understanding of the environment changes. In general, many applications have solved the VPR task as an IR problem [63], in which images of the same location indicated in a query image are recovered from a large geo-tagged image dataset. Creating reliable image representations is a major focus of image retrieval, as this allows the system to acquire a metric (such as the Euclidean distance of feature vectors) to determine whether or not the images are a good match. Another challenge is that finding the geolocation in a large dataset requires a fast image representation. A high level of recognition accuracy, for instance, requires discriminating between numerous objects, such as cars, trees, and people in a street-level image. To achieve efficient visual place recognition, retrieval methods will need to overcome these obstacles.

Recent studies have shown that extremely high-dimensional descriptors, like the Vector of Locally Aggregated Descriptors (VLAD), can achieve high retrieval accuracy over a large- scale image dataset [57, 63, 8, 50]. VLAD is used to create visual dictionaries and vectors to represent features in images. VLAD defines an image by comparing its local feature descriptors to a codebook that has already been calculated. Typically, k-means clustering of the descriptors produces a visual codebook. However, visual features have a dimensionality that is not simple, and computing sample distances in a large image collection is difficult. We focus on creating an accurate image retrieval technique with reasonable calculation costs and a fast search time.

In this thesis, we also propose a method of learning image representations with the goal of enhancing VPR. To compute the similarity between vectors in VLAD, we randomly apply mini-batch k-means. Because binarized VLAD permits fast Hamming distance computation and light storage of visual descriptors, we compress VLAD to binary aggregated descriptors to improve search speed. Our research shows that these enhancements improve visual place retrieval performance over current state-of-the-art techniques.

1.2.2 Informative Feature Selection for Visual Place Recognition

Since the environment's appearance can shift suddenly and unexpectedly, scientists have spent a lot of time studying how to design VPR systems that can keep working without breaking down. Since it was found that not all image content is useful for VPR tasks, a lot of research has been done in the area of feature selection. When it comes to remembering familiar landmarks, a building's window is far superior to a t-shirt or a set of wheels. Therefore, rather than relying on the entirety of an image's content to signify its meaning, Kim et al [63] proposed that visual representations intelligently highlight useful regions to boost similarity to the relevant images while suppressing the regions that cause overlap with irrelevant images. Alternative methods have been proposed to selectively target local regions based on their particular features [65, 55, 8]. However, all these methods focus on improving the VPR performance in terms of robustness. Unfortunately, all the existing methods require time-consuming computations, such as querying each database image to identify good or informative features, that are not sufficiently scalable and are thus less practical. While successfully matching different places under changing viewpoints and conditions remains the top requirement of a VPR system, computational and storage needs should also be considered to achieve the practical deployment of a VPR technique. In this thesis, we propose using a hybrid feature set, which is a robust image representation method to distinguish between relevant and non-relevant features in large datasets and challenging environments. We then reduce the number of hybrid features during similarity measurements by predicting good features in the offline phase. Finally, instead of forcing the algorithm to learn relevant and irrelevant images, we propose a method that can predict unique features by learning non-relevant features in a data-driven way. According to our research, these contributions improve visual place recognition performance when compared to baseline techniques.

1.3 Thesis Contributions

This dissertation has four contributions that have advanced the state of the art in visual place retrieval and recognition. Our contributions to this thesis are outlined in more detail below.

1- We use a geotagged image dataset to investigate VPR for an image query. Low-cost image retrieval methods represent the image contents as feature vectors. The Vector of Locally Aggregated Descriptors (VLAD) is a type of low-cost method that can be used for visual place recognition. VLAD characterizes an image by comparing its neighbourhood features to a precomputed dictionary. Descriptors are typically clustered using k-means to create a visual codebook. Unfortunately, visual features have a non-trivial dimensionality, and computing sample distances in a large image collection is challenging. To create an efficient, low-cost image retrieval method, we suggest using mini-batch k-means clustering to generate VLAD descriptors (MBVLAD). The proposed MBVLAD methodology achieves higher levels of retrieval accuracy than existing methods.

2- Our proposed method for representing images combines an aggregated binary descriptor – Oriented FAST and Rotated BRIEF (ORB) – with an MBVLAD descriptor. When working with a large database, however, the time and resources required to extract and compare pairwise the local descriptors create a bottleneck, limiting the efficiency of local feature matching between images. We define binary local features for the aggregation method to reduce the cost of extracting, representing, and matching local visual descriptors, thereby increasing the efficiency of local features. The search accuracy (mAP) and search time (s) we measured in our experiments demonstrate that our ORB-MBVLAD is significantly faster than other state-of-the-art methods while maintaining high levels of accuracy.

3- Long-term environmental change is one of the biggest challenges for VPR systems. Identifying areas of focus is a useful strategy for tackling this issue. Feature selection is an active area of study; however, finding a strong image representation capable of discriminating unique features is challenging. Most of the features used are hand-crafted, and they perform particularly well in visual localization and location recognition tasks. However, it is challenging to select what sort of features should be utilized to characterize locations, since hand-crafted representation demonstrates superior awareness of a place in rotational perspectives but struggles with particular surroundings (i.e., trees, buildings, or mountains). As deep learning networks advance rapidly, it is becoming clear that learned features outperform hand-crafted features in place recognition tasks. Unfortunately, this is not a good use case for deep learning representations for selecting features in an outdoor setting, where conditions can shift when going from day to night. By fusing traditional and deep learning approaches, we propose a hybrid feature method for representing images with greater robustness. Furthermore, we select useful features using our hybrid feature in a data-driven way. A comprehensive performance comparison of different representation methods for selecting features for VPR was conducted. Our hybrid feature shows a significant improvement compared with selecting features using a single method for a VPR task.

4- In this thesis, we offer a unique VPR algorithm that is robust against environmental changes. Obviously, outdoor environments with moving objects like automobiles, people, and so on might confuse location-based systems, but throwing out features that are derived from moving objects in an image might be risky. At the same time, using such features increases the computational cost of calculating the similarity between the input image and the reference images. We suggest learning to recognize confused features from data and then using that knowledge to predict relevant features in a query image before performing geo-localization. We demonstrate both quantitative and qualitative improvements over previous state-of-the-art methods of VPR.

1.4 Thesis Outline

The works undertaken in this thesis is structured into seven chapters:

The work undertaken in this thesis is structured into seven chapters: In Chapter 2, we explore the background and architecture of the VPR system. The most current advancements in VPR and retrieval are discussed, with an emphasis on place recognition in continuous operations. After this, the problems with current VPR systems are summed up. This chapter provides the VPR and retrieval datasets as well as assessment criteria for both systems. Chapter 3 introduces the first contribution of this thesis, which is MBVLAD for visual place retrieval. In this chapter, the solution to the retrieval system is proposed. To represent images in an efficient way, the similarity distance between two vectors was learned and described in a paper published at ISSC in 2019, which was named the best student paper [5].

In Chapter 4, the second contribution is presented. An efficient visual place retrieval system using Google Street View is proposed, and most of this chapter was published at IMVIP in 2020 [2].

Chapter 5 provides a third contribution. A method for predicting useful features is provided. Also, a new image representation method that is a hybrid feature is presented. This work was published at ICDIP in 2022 [4].

Chapter 6 presents an efficient VPR using useful features. In particular, we provide a novel method that uses predicted features from relevant images to analyse the input image. The data-driven way to generate features and learn classifiers is presented. This is the last contribution to this thesis. Most of this chapter was accepted at CC-GIV in 2022 [3].

In Chapter 7, we conclude our main contributions introduced in this thesis and discuss the potential directions to be explored in future work.

Finally, the appendices provide a number of supplementary results generated by the methods proposed in this thesis.

1.5 List of Publications

A number of paper publications took place during the preparation of this thesis:

- **Aljuaidi**, R. , and Su, J. , and Dahyot, R (2019) Mini-Batch VLAD for Visual Place Retrieval. 2019 30th Irish Signals and Systems Conference (ISSC). **Awarded Best Student Paper at ISSC 2019**. https://doi.org/10.1109/ISSC.2019.8904931.

- **Aljuaidi**, R. , and Dahyot, R. (2020) Efficient Visual Place Retrieval System Using Google Street View. 2020 Irish Machine Vision and Image Processing (IMVIP). http://research.thea.ie/handle/20.500.12065/3429

- **Aljuaidi**, R. , and Manzke, M. (2022) Predicting Good Features Using A Hybrid Feature For Visual Geolocation System. 2022 14th International Conference on Digital Image Processing (ICDIP). https://doi.org/10.1117/12.2645302.

- **Aljuaidi**, R. , and Manzke, M. (2022) . VPR. 5th International Conference on Computer Graphics, Images and Visualisation (CCGIV). https://doi.org/10.1145/3569966.3570105.

The following is a publication that is related to the thesis but not included in the thesis:

- **Aljuaidi**, R. (2022) Image Geolocation System for Road Signs Main-

tenance. 2022 19th International Operations and Maintenance Conference in the Arab Countries (OMAINTEC).

Chapter 2

Background and Related Works

As a first step, this chapter introduces an overview of visual place recognition strategies. After that, we show several methods of using images to describe a place. Then, image representation methods for visual place retrieval are described. Following that, we provide efforts of similar kind in the field of visual place recognition. We then go on to talk about a variety of research projects that address the impact of changing environments on image retrieval and visual place recognition. At long last, we cover the benchmark datasets for VPR and retrieval and their respective assessment techniques.

2.1 An Overview of visual place recognition (VPR)

Because of the decreasing costs of cameras and the amount of sensor data available, place recognition for visual localization is becoming more popular in real-world applications [20, 25, 90, 128]. In this context, each place may be represented by an image or series of images, allowing a human, robot, or vehicle to locate itself by remembering a known location from memory. As a result, place recognition-based techniques in an outdoor environment must be robust and efficient under environmental changes to provide accurate location information. Solving the problem of where an image was taken under environmental changes has emerged as a major research challenge [50, 8, 63, 20]. Having to cope with fleeting or regionally pervasive visual features is made more complicated by the fact that the look of a place might vary drastically during the day [55]. The basic process of a location recognition system is shown in Figure 2.1. In order to understand each stage, please refer to the descriptions below.

- **Inputs:** Images and videos are the primary sources of information for the whole system. Data preprocessing also includes transforming raw data (such as a set of feature descriptors or an entire image) into a more usable format for description or storage.

- **Place describing:** Places need to be defined in a manner that makes them easy to remember and find again. There are two main categories of visual place description methods: those that describe just the selected sections of a scene and those that describe the full area. With the goal of improving retrieval and recognition systems,

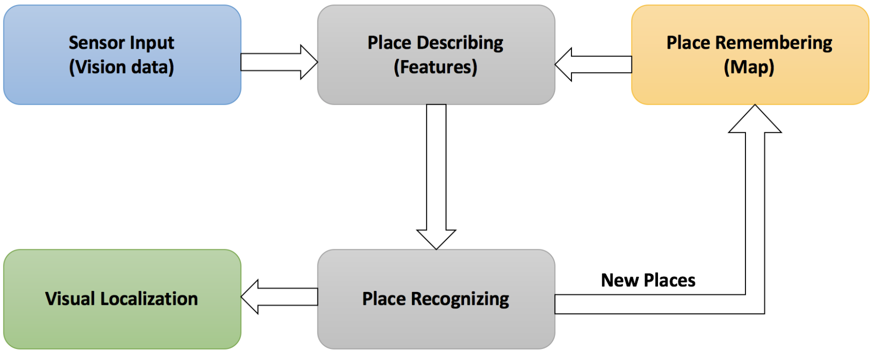


Figure 2.1: Five main parts make up this overall strategy for a visual place recognition system. The place description subsystem analyzes incoming visual information. The module that can remember where you've been stores the geotagged images. The current visual data is compared with the database to see whether it matches any of the recorded locations. Using the geotagged information from a previously visited location, the performance is able to precisely localize itself by linking the location of the place to a returned or matched image.

this thesis concentrates on the place describing phase. Given that it is possible to extract features from and represent images during computation in a manner analogous to a map, describing a place is the most fundamental method of describing a particular environment. In this method, image retrieval techniques are used for the primary purpose of recognizing locations based on their outward appearance.

- **Place remembering:** In order to compare and get the extracted descriptors, a location recognition system has to refer to a map. Recent methods may be roughly categorized as either those that rely on topological maps, metric maps, or geolocation databases. How a location is remembered changes depending on the system's intended use. For the purposes of this thesis, we use a database of geotagged images.

- **Place recognition:** Matching new data with old information is what happens during place recognition. Last but not least, the goal of place recognition is to determine whether a certain place has been visited before. Everybody knows that if two locations sound the same, they must have been shot in the same place. Therefore, the fundamental goal of any location recognition system is to create a belief distribution by matching visual input with historical information. You may either use a single image or a collection of images to recognize a location. In this thesis, we investigate the problem of localization from a single image.

- **Out-put:** The output is the geo-tagged image from previously visited areas to pinpoint its precise location during the place recognition process.

2.2 Visual Place Recognition Based on Content Based Image Retrieval

There are two main categories when it comes to recognizing places in street-level input images: those that rely on image retrieval and those that rely on 3D structures. Image retrieval-based methods estimate the geo-location of a query image by comparing it to reference images portraying the same place [63, 55, 8]. Our thesis fits within this group. Combining these methods with others for visual location detection, such as Seq-SLAM citation [22], makes for a powerful tool. Another approach that may be used to estimate the location is to have users vote on geo-location tags associated with local features in order to get a more precise estimate than the location of the most comparable image in the database [146]. This is a computationally expensive scenario because each local feature in the query image is checked against the database for its closest nearby local feature. It is important to keep in mind that none of these algorithms can determine the precise camera posture of the question on their own.

The 3D structure-based methods frame the issue as a 2D-to-3D registration task using a 3D model built from images in a database [108, 49, 73]. These algorithms can do a full camera pose estimation based on the query image. However, they are only practical in places where there is a great concentration of reference images and they need constant upkeep. While certain Convolutional Neural Networks CNNs can be trained to anticipate the camera pose from an input image [59], this approach has the same high maintenance cost as other approaches that encode 3D structure implicitly. There have been attempts to combine the two methods in order to recreate the 6DOF posture from the retrieved reference images. Using a map associated with the images in the database, [106] one of the first efforts calculates the pose. To determine the camera's position, Zhang and Kosecka [148] utilize a pair of reference images. In order to accurately forecast the pose of the query image, Sattler et al. reference [110] recently suggested building a local 3D model from the selected set of returned images.

The majority of the time, we tackle the problem of visual place recognition as an image retrieval task. The success of this idea hinges on our ability to create place-based visual representations. In this subsection, we will quickly review the hand-crafted representations used for this purpose before the advent of CNNs. As a city-scale image retrieval task, we investigate the problem of visual place recognition. Other than the obvious photometric and geometric differences between the query and reference images, the significant visual overlap with irrelevant images caused by ubiquitous visual features like pedestrians, cars, billboards, and trees also presents a significant challenge to visual place retrieval. Those massive attributes may have an impact on the time and money required to perform computer searches and analyses. Adaptive image representations are one way we've been able to tackle these problems (chapters 3 and 4). To further combat computational expense during image representation, we use a data-driven strategy to reduce the total number of features (Chapters 5 and 6).

The fundamental pipeline of a visual place recognition system is shown in Figure 2.2. Initial steps include transforming a raw image of the place under investigation into a more quantifiable mathematical representation (usually a feature vector). The feature vector is then used to calculate the degree of similarity between the images in the database and the ones in the query. Two images' similarity score may be used to determine whether they were taken in the same place. When the score is over a certain threshold, the two images are deemed to be a match. A significant part of VPR is the process of representing images. It is important to have a strong visual representation that can accurately differentiate the relevant aspects when describing a place. We provide image representation strategies for place retrieval in the next section.

2.3 Image Representation Methods For Visual Place Retrieval

In this thesis, we focus on improving VPR and retrieval systems. Image representation is an important step in describing a place. As we

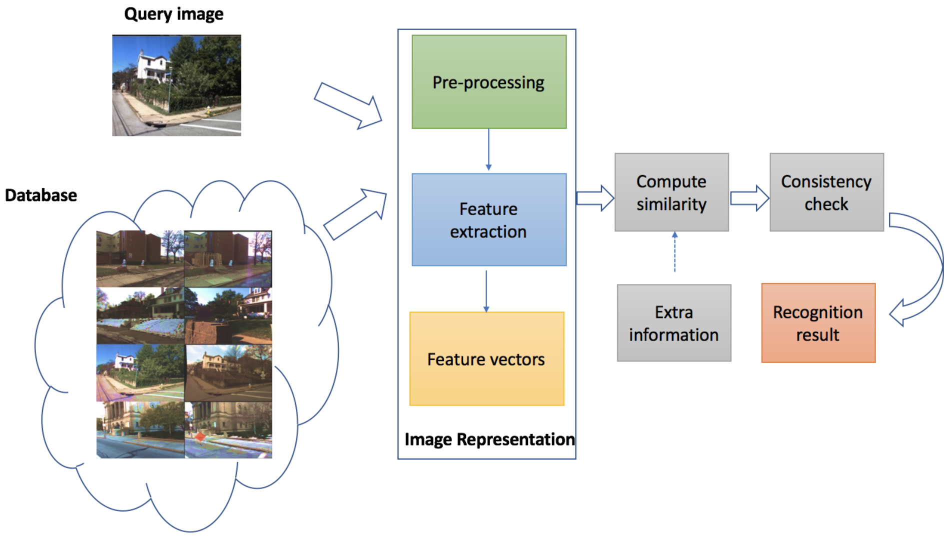


Figure 2.2: A basic pipeline of visual place recognition [150].

mentioned above, visual place recognition is the first step and should describe a place in order to recognize it correctly. Methods of visual representation are used in order to accomplish this goal. Image description, information retrieval, localization, 3D reconstruction, machine learning, and many other fields of research all have a stake in the study of image representation.

The unique features of each image are codified by identifying its salient aspects. These features are then utilized to build feature vectors or image descriptors. In other words, the essential elements of each image are extracted and combined into a single composite. Afterwards, a comparison is made between the two image descriptors, often using the L2 norm or the Euclidean distance method. Scores close to one another reflect the degree of similarity between the two input images. When the number of references to be compared is manageable, it is also common practice to use the "closest neighbor" approach to conduct a comprehensive comparison of the whole database. More sophisticated approaches to data pre-processing and searching are required for larger-scale applications.

Image representation approaches contain two methods: traditional methods and method-based learning. The traditional approaches are also defined as hand-crafted features due to the characteristics of feature selection. Elements such as corners, borders, blobs, and ridges give geometric information, whereas features such as color, pixel intensity, texture, and contour provide low-level visual details. In the following, we will provide a thorough presentation of each category.

2.3.1 Traditional Image Representation Methods For Place Retrieval

Methods Based on Local Features

By making use of salient points or regions of interest, local-based approaches represent an image. Scale-Invariant Feature Transforms (SIFT) [77] and Speeded Up Robust Features (SURF) [11] are the methods dominating the early state. In 2011, Oriented FAST and Rotated BRIEF (ORB) were introduced [107]. ORB builds on the well-known FAST keypoint detector and the BRIEF descriptor. Both of these techniques are attractive because of their good performance and low cost. Also, a learning method for relating BRIEF features under rotational invariance leads to better performance in nearest-neighbor applications.

Bag of Words (BoW) techniques [118] are sometimes combined with a visual dictionary to improve accuracy [117]. A visual vocabulary or visual codebook (dictionary) of local feature repetitions is created as part of the first step of the procedure. One common method of quantization is the grouping of visual words with the use of an algorithm like k-means clustering. A Vector of Locally Aggregated Descriptor (VLAD) [57] is a variant of a Bag of Words (BoW) [118, 92] that adds a

residual to each descriptor based on its cluster membership. In other words, we may calculate the total of the differences between each cluster's assigned descriptors and the cluster's centroid by matching each descriptor to the cluster that has the closest instance of that descriptor. The aggregated description for VPR is shown in 2.4.
The probabilistic FAB-MAP approach [23, 24] is the most well-known in vision-based place recognition among the systems utilizing local-based features; the system recorded the scene using more than 100,000 visual words and utilized them to sequentially track the current place. For the purpose of closing the loop, researchers turned to the place recognition technique, which basically means that our system recognized previously visited locations based on the similarities between the recorded visual words. Each had its own index for rarity, which added to the overall scene's uniqueness. Viewing invariance comes from local features in most cases [78].

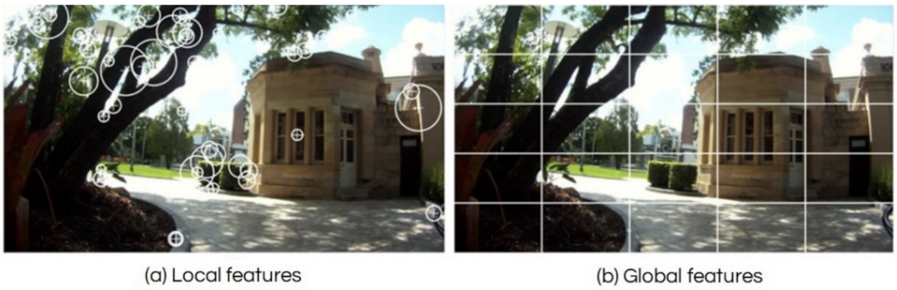


Figure 2.3: An example of local and global features methods. The image is taken from [78].

Methods Based on Global Features

Globally-oriented strategies apply the methodologies to the whole picture rather than concentrating on specific regions. Histograms may be used to describe the whole image, as in the case of color histograms. In this scenario, the bins represent the dimensions of a

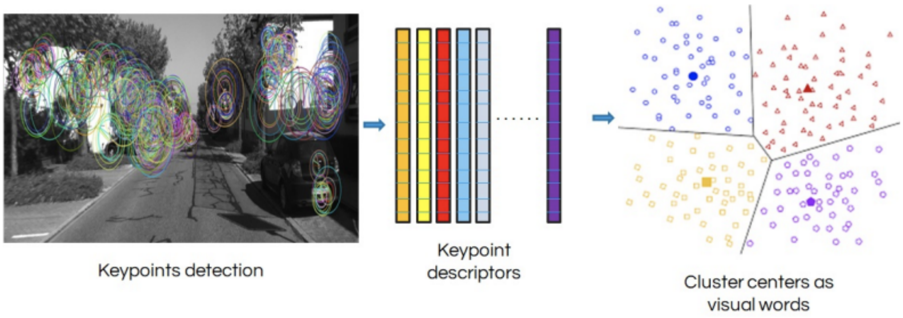


Figure 2.4: An example of aggregated with clustering.

color space, and each pixel's contribution to the histogram is determined by its value within that space. Another popular representation is the gradient histogram, which shows how gradient orientations are distributed throughout a picture. A well-known example that has been successfully used for person identification in photos is the Histograms of Oriented Gradients (HOG) descriptor [26]. However, they are more resistant to photometric changes such as shifts in light due to their holistic nature. An additional widely used descriptor, GIST, was first introduced by Oliva and Torralba [96]. One of the best-known global-based approaches is the GIST descriptor, which is a low-dimensional holistic scene descriptor using the spatial envelope features of a spectral representation. Compared to localized methods, globalized methods need less storage and processing power. Many works have used these properties for location-based applications like large-scale location matching [31, 88, 89] and positioning and navigation [134, 99, 116, 105].

Methods Based on Local and Global Features Combination

One alternative is to utilize a global descriptor to quickly narrow down search results to relevant images before moving on to a more precise method, such as matching local characteristics, to confirm the relationship and ensure reliability. Omnidirectional cameras are used in the localization technique presented by Goedemé et al. [44], which involves the extraction of vertical column segments from each captured picture and their subsequent description using 10 distinct descriptors. The localization procedure takes advantage of these neighborhood descriptors, which have been clustered, by inserting them into a kd-tree structure. The incoming query image is processed using a global computation of the same local descriptors applied to the vertical structures, which is then utilized to quickly obtain candidates for the loop. The next step in ensuring an accurate image match is to apply a matching distance based on the column segments to the image and each of the contenders. In the work [79], it is suggested to combine the local visual characteristics FAST (Features from Accelerated Segment Test) and CSLBP to create a robust and real-time visual location identification system (Complete Center-symmetric Local Binary Patterns). To implement omnidirectional vision-based location identification for mobile robots, we employ bag-of-features and support vector machines to analyze the provided key features. According to the findings of the experiments, the robot is capable of accurate real-time place recognition with a high categorization rate. By continually creating the convex hull from the retrieved SURF features and measuring the relative magnitude between these features that construct the convex hull, Wang and Lin offer a combined local and global descriptor for omnidirectional pictures dubbed Hull Census Transform (HCT) [133]. Then, we utilize this representation to identify scene changes. Wang and Yagi [132] suggested a method for place recognition that used edges, local features, and color histograms all together. The Harris detector is utilized to produce edges and interest points, and SIFT [77] is used to describe interest points, enabling a fully calculated, integrated image description process.

To sum up, local features have several advantages over global descriptors. To begin, features may be used for more than just recognizing particular objects or places. Making maps, joining images, and redefining places all fall under the category of "place recognition." They may be easily combined with metric information to enhance localization tools. Additionally, keypoint-based methods are not as dependent on camera pose since they are robust to changes in geometry. On the other hand, global descriptions tend to be very subjective. High photometric and geometric invariance may be achieved by a compromise reached by applying global descriptors to image segments rather than the complete image. But when the dataset is large or the number of extracted features is enormous, local and global descriptors both hit a wall. The high computational and processing costs have a negative effect on efficiency. The third and fourth chapters are dedicated to creating a way of representing images utilizing aggregated local descriptors.

2.3.2 Methods Using Deep Learning

The primary difference between traditional and learning-based methods is that the latter rely on the system's own internal mathematical models developed over time for autonomous prediction and decision-making. To learn, one must first be able to automatically recognize and adapt to new situations based on the patterns they see in the data. There are two main types of learning-based techniques: unsupervised learning (clustering) and supervised learning (classification). The former category includes popular techniques like k-means clustering and the Gaussian Mixture Model (GMM), which require neither previous knowledge nor labeled data. Its goal is to discover latent relationships among unlabeled data and organize it into meaningful chunks. The latter uses Support Vector Machine (SVM) [46] extensively to work on pre-defined picture classifications.

The convolutional neural network (CNN) is the most well-known deep-learning approach, particularly for image-based problems. This model was introduced in the 1990s, but it was not widely used until the AlexNet model [67] was introduced, which set a new standard for object classification over millions of data points. Two common techniques are used to apply CNN to image retrieval and matching tasks: (i) generating similarity functions and (ii) extracting features.

The AlexNet model [67] uses the Siamese model to discover the relationship between the pairs of images it is given. CNN layer outputs serve as descriptions for the resulting images. The AlexNet architecture is shown in Figure 2.5. Three fully connected layers and five convolutional layers (conv1, conv2, conv3, conv4, and conv5) make up the model (fc6, fc7, and fc8). Also, AlexNet has 600 million parameters and 650,000 neurons. Because of this, it is challenging to understand their learning. It's possible that further research is needed on this issue. The field of visual place recognition is no exception to the widespread adoption of CNN models.

With fewer parameters, SqueezeNet [54] (a compact CNN design) achieves AlexNet-level accuracy [67] on ImageNet [28]. The SqueezeNet architecture has been used in many applications, such as real-time vehicle make-and-model recognition [70], searches for semantic segmentation [113], and image classification [54]. The SqueezeNet design employs tactics like downsampling in the last few layers of the network to keep activation maps for the convolutional layers wide while decreasing the number of input channels from 3x3 filters to 1x1

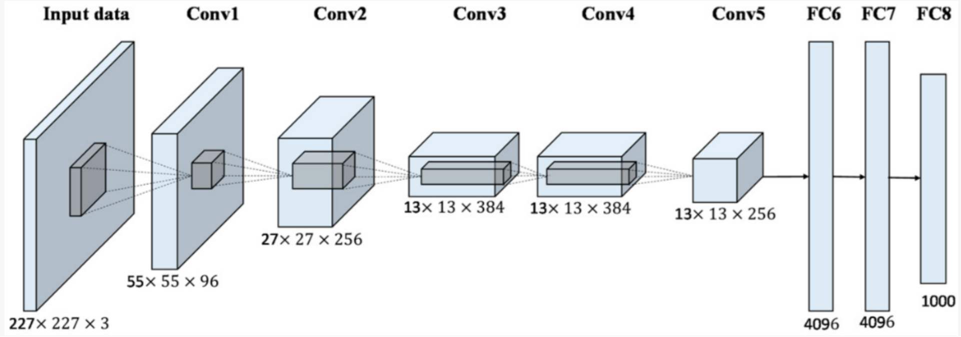


Figure 2.5: AlexNet's [67] internal structure. Three fully connected layers and five convolutional layers (conv1, conv2, conv3, conv4, and conv5) make up the model (fc6, fc7, and fc8). That picture comes from [47].

filters. An example of this is the Fire module, which employs a combination of 1x1 and 3x3 filters in the expand layer and a series of 1x1 filters in the squeeze convolution layer. There are around 1,248,424 parameters and 4.85 MB of model space in the Fire module of the SqueezeNet architecture (shown in 2.7). The architectural dimensions are presented in Table 2.1.

Table 2.1: SqueezeNet architectural dimensions

Layer	1x1	1x1	3x3	Filter size	Output shape
Input					224x224x3
Conv 1				7x7 2	111x111x96
Maxpool				3x3 2	55x55x96
Fire 2	16	64	64		55x55x128
Fire 3	16	64	64		55x55x128
Fire 4	32	128	128		55x55x256
Maxpool				3x3 2	27x27x256
Fire 5	32	128	128		27x27x256
Fire 6	48	192	192		27x27x384
Fire 7	48	192	192		27x27x384
Fire 8	64	256	256		27x27x512
Maxpool				3x3 2	13x13x512
Fire 9	64	256	256		13x13x512
Conv 10				1x1 1	13x13x1000
Global avgpool				13x13 1	1x1x1000
Softmax					

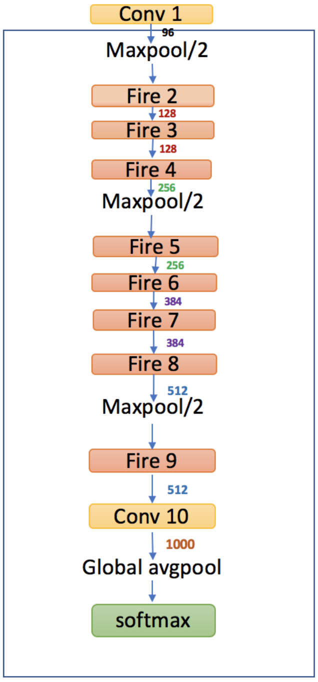


Figure 2.6: CNN framework based on the SqueezeNet. A single convolution layer (conv1), followed by eight Fire modules (fire2-9), and a final convolution layer make up SqueezeNet (conv10).

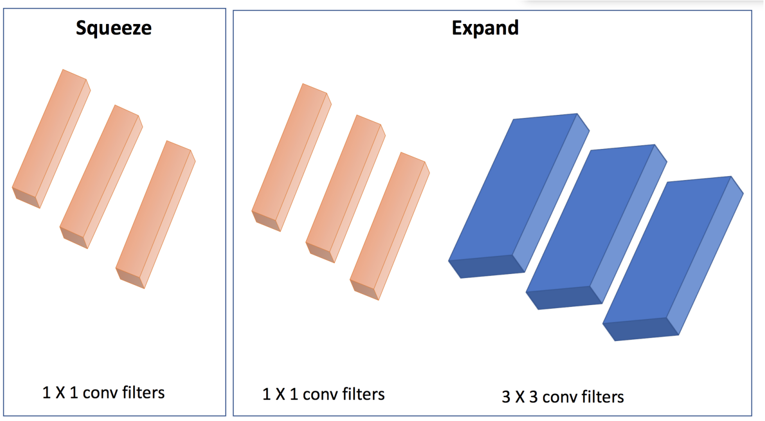


Figure 2.7: The structure of the Fire module's convolution filters. In place of 3x3 filters, there are now just 1x1 filters. There are only so many convolution filters available, and most of them are 1x1 filters since they need nine times fewer parameters.

Though AlexNet and SequeezNet perform better on feature extraction and classification tasks, they are not a good fit for VPR and retrieval. Lighting and appearance variations might throw off visual location identification techniques that rely on manually generated elements. Their success in difficult settings is highly dependent on the robustness of such descriptions against changes in how things are seen. To improve the feature extraction process, we use deep learning in Chapter 5.

The research article [22] presents a ConvNets-based approach to place recognition by integrating the robust features learned by ConvNets with a spatial and sequential filter. Sünderhauf et al. [122] to provide an innovative technique for recognizing locations based on recent advancements in object identification technology and convolutional visual features. The suggested NetVLAD architecture creates a robust image descriptor for the aim of visual place recognition, as shown in Arandjelovi'c et al. [8]. The authors of the study [8] create a convolutional neural network architecture that can be trained from beginning to end. The core of this design is a novel, generic VLAD layer (NetVLAD). The layer may be easily plugged into any preexisting convolutional neural network design and trained using standard backpropagation methods. The CNN design using the NetVLAD layer is shown in Figure 2.8. On the other hand, NetVLAD necessitates picking out both positive and negative samples for each training picture. important for training a model with contrastive and triplet losses.

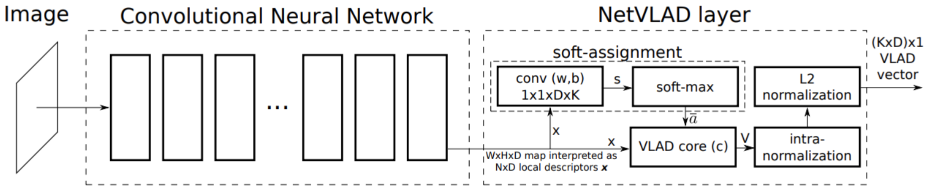


Figure 2.8: CNN architecture with the NetVLAD [8].

2.4 Image Representation by Aggregation of Local Features for Image Retrieval

A suitable mathematical description of each image is necessary in order to determine if two images are comparable in terms of visual content or whether they include the same object. In the last section, we outline several of the most well-known methods for converting an input image into a numerical descriptor.

Content-Based Image Due to the difficulty of local feature extraction and matching, retrieval based on local features is computationally costly. On the one hand, the cost of extracting [107, 38], expressing, and comparing local visual descriptors has been greatly reduced by the newly suggested binary local features. But aggregation methods allow for compressing all the image's obtained features into a single meaningful descriptor, which greatly improves the speed and scalability of image search. Recently, only a handful of studies have merged these two lines of inquiry by providing aggregation methods for binary local features, allowing users to reap the benefits of both approaches. Our focus here is on binary local features, and we will go through the most common aggregation techniques and how they work.

2.4.1 The Bag of (Visual) Words and Binary Local Features

Each image is represented as a collection of visual words in the Bag of (Visual) Words (BoW) [118] by classifying the image's local descriptors according to a standard visual vocabulary. The visual vocabulary is constructed by grouping the dataset's local descriptors, maybe using k-means. The cluster centers, or centroids, represent the visual words of the vocabulary and are used to quantify the local descriptors that were acquired from the images. Each local descriptor of an image is assigned to its closest centroid, and the image itself is represented by a histogram of the frequency with which the visual terms appear. The retrieval process makes use of text retrieval methods, except that visual terms are used instead of text words and the query image is processed as a disjunctive term query. The cosine similarity measure is often used in combination with a word weighting system, such as phrase frequency-inverse document frequency, to evaluate the degree of similarity between any two images (tf-idf).

In order to include binary features in the BoW approach, a cluster method is necessary that can deal with binary strings and Hamming distance. The k-medoids [58] work well for this, but it takes a lot of processing power to calculate the whole distance matrix between the elements of each cluster. It was proposed in [46] that a k-majority voting method be used to analyze a sequence of binary vectors and find a collection of acceptable centroids that would act as the BoW model's visual words. To better match the binary features, the [151] representation uses the median operation instead of the mean and the Hamming distance rather than the Euclidean distance.

2.4.2 Vector of Locally Aggregated Descriptors and Binary Local Features

Since binary vectors are a particular instance of real-valued vectors, the VLAD technique [57] may be easily applied to binary local descriptors. The k-means method may be used to construct the visual vocabulary in this fashion, and the difference between the centroids and the descriptors can be accumulated as usual. The BVLAD image signature, a variant of the VLAD image signature optimized for use with binary features, was also developed using this approach as well [130]. The BVLAD is a VLAD that has been binarized (by thresholding) after being generated using power-law, intra-normalization, L2 normalization, and multiple PCA.

A visual vocabulary may be built by associating each binary descriptor with the nearest visual word using the Hamming distance and a variety of binary-cluster algorithms, including k-medoids and k-majority, in a manner similar to BoW [38]. Calculating the residual vectors using binary centroids may provide less informative results. Using mini-batch k-means [5], we propose to study this issue in Chapter 4.

2.5 Challenges and Key Strategies in Real-world Place Recognition

The following difficulties may be encountered by sensors in real-world application tasks such as navigation: These include, but are not limited to, the following: These include, but are not limited to, the following: 1) varying visual appearances due to time fluctuations; 2) various perspective variances for the same places; 3) discovering new, unknown areas; 4) implications for efficiency and robustness when applied to real-world settings. In this thesis, we use a retrieval task to address the efficiency and robustness challenges that arise when dealing with changes in visual appearance.

2.5.1 Appearance Change

A significant challenge in the field of place recognition is the phenomenon known as "perceptual aliasing," which occurs when two or more locations provide visually identical data. There are two kinds of visual changes that affect long-term place recognition:

- Conditional changes, changes in appearance due to factors like lighting, weather, and the passage of time, are included in the category of "conditional changes." Over time, this kind of change will have a mostly visual impact.

- Structural changes, short-term and long-term navigational changes in the structure, including dynamic items, geometric modifications, and land form changes. Vision and LiDAR sensors are vulnerable to interference, while radar is safe due to its ability to observe at a low frequency. [53].

As mentioned in [150], place modeling, which employs conditional invariant features for stable place recognition, and belief generation, which estimates place similarity based on the sequence of observations, are the two primary kinds of ways to cope with the aforementioned appearance changes. As a result of recent advances in computer vision [145, 107, 127], deep learning [115], and adversarial learning [45, 153], place modeling-based PR methods [60, 103, 50] have emerged. Tsintotas et al. present a vote technique to identify probable loop closures inside a distributed database and a training process to capture recurring scale-restrictive features in [127]. For effective relocalization, Merrill et al. [83] integrate information from many visual modalities to derive rich place recognition elements. A region-based VLAD feature aggregation module is provided by Khaliq et al. [60]. This module makes use of the pretrained AlexNet [67]. The accomplishments of region-based visual object recognition inspired the development of AlexNet for trustworthy local region feature extraction [67]. When it comes to improving the precision with which computers can identify specific locations, Hausler et al. [50], a multi-scale patch feature fusion approach. The local features may be aggregated across the feature-space grid using Hausler's technique, which is resilient to variations in environmental conditions. The aforementioned approaches rely heavily on the pretrained model using existing datasets, which might restrict their performance. To achieve competitive recognition performance without training, by combining the traditional HOG [26] descriptor with the regional feature extraction and convolution matching technique, Zaffar et al. [145] provide a training-free VPR solution. For the VPR task, Piasco et al. mention [103] as a new modality that, if used correctly, may infer depth prediction from the visual inputs; this modality is the visual CNN features. This approach, however, depends on the presence of coupled image-depth data. As a result, the quality of the dataset and the level of detail in the images greatly affect localization performance and generalizability.

The most well-known sequence matching strategy for belief generation-based PR is SeqSLAM, which was initially reported by Milford et al. in [85]. SeqSLAM finds the best matches by aligning a pair of reference and query sequences, and it can accurately capture the continuous geometric similarities under conditional changes using even the most conventional visual features, in contrast to traditional single-frame-based place recognition methods like FAB-MAP [23] and Bag-of-Words [40]. As of 2012, a sizable body of work has used the SeqSLAM method in order to enhance the precision of location identification [120, 10, 76, 140, 142, 19]. Stone et al.[120] combine the SeqSLAM matching mechanism with skyline segments, which exhibit improved condition-invariant behavior over rough pathways, to stabilize place recognition ability on challenging terrain in changing environments. By adopting the Approximate Nearest Neighbor (ANN) to greedily scan the sequence in place of the standard burst-force searching, the fast version of SeqSLAM developed by Siam et al. [114] may drastically reduce the searching time without compromising the localization accuracy. As this approach requires a precise estimate at the outset, it can't be used for large-scale global re-localization initiatives. Yin introduces a global place recognition method based on multi-resolution sampling, that can be used for both visual place recognition (VPR) [140] and Linguistic Place Recognition (LPR)[142]. In order to achieve hierarchically global re-localization, Yin's method combines coarse-to-fine re-sampling with a particle-filter. This method can strike a good balance between matching accuracy and efficiency, and it can aid in providing near real-time global localization capability in long-term navigation. Bampis et al. [10] propose a sequence bag-of-words with a unique temporal consistency filter that can gain from sequence matching while still maintaining real-time performance on a tablet, as opposed to the burst-force searching through the difference matrix used in SeqSLAM. A combination of a compact and sparse neural network (FlyNet) with a continuous attractor neural network (CANN) to capture the sequence of observations has been shown to outperform SeqSLAM [19]. Also employing a temporal convolutional network, Garg et al. [41] offer a global sequence descriptor that, when combined with local sequence matching, allows for a hybrid location recognition system.

2.5.2 Viewpoint Difference

One of the major obstacles to place recognition is that changes in viewpoint may impair the identification capacity for all sensor modalities. Observations made at the same location over time may reveal different patterns, depending on the perspective. For over a decade, researchers have studied classic place recognition techniques like bag-of-words [118] and the Vector of Locally Aggregated Descriptors (VLAD) [57] to find ways to make the perspective difference more robust. However, the above techniques are incapable of handling ad hoc variations in perspective or opposite orientations. As a result of perspective effects, position-based place identification will consider the resulting shifts in distribution to represent whole new locations. The neural networks provided by Garg et al. [42] are semantically aware, meaning that they can identify the same location when seen from different perspectives. But this technique only works for reverse-direction detection, whereas indoor or outdoor navigation often encounters arbitrary variances in the real world. The polar context projection used by Kim et al. [62, 61] provides a rotation-invariant descriptor, and the combination of rotation-invariant network structures is proposed by Li et al. [71] for a rotation-invariant place descriptor. Using the orientation-equivalent feature of spherical harmonics, Yin et al. [142] map LiDAR or 360-degree visual inputs to the spherical projections, which may be used for orientation-invariant location identification. To take advantage of the translation-invariant feature of top-down inputs and the orientation-invariant property of a spherical view, Yin et al. [144] present a multi-perspective fusion-based LPR based on dense local maps. The localization accuracy for the noisy radar inputs is enhanced by the orientation-invariant RPR approach presented by Suaftescu et al. in [39], which employs cylindrical convolutions, anti-aliasing blurring, and azimuth-wise max-pooling.

2.5.3 Generalization Ability

The capacity for generalization is indicative of the ability to recognize locations in settings that are not immediately apparent. Since the same place can be shown in different ways, there is no limit to the different ways the above place datasets can be put together. Even if it were possible, it would be hard to get all of them at once. To accomplish widespread visual re-localization using conventional techniques, FAB-MAP [23] constructs a Bag-of-Words (BoW) architecture. In order to obtain matched images faster, iBoW-LCD uses an incremental BoW technique based on binary descriptors. To identify loop closures incrementally, An et al. provide FILD++ [6], which uses a hierarchical small-world network. However, the following non-learning-based approaches have very limited generalization capacity, and robust performance often requires precise parameter fine-tuning. Learning-based place feature extraction approaches have seen increased interest in recent years due to the advancement of deep learning-based feature extraction [115, 52] and attention mechanisms [131]. For the VPR challenge, Khaliq et al. [60] integrated the area of CNN features with a differentiable NetVLAD [8] layer to facilitate generalization. Zhang et al. [149] propose a Point Contextual Attention Network (PCAN) based on the attention mechanism to enforce differential networks by giving more weight to task-relevant information to improve the robustness of LPR approaches. To enhance localization performance when faced with occlusion and divergent viewpoints, new research by Kong et al. [66] presents semantic graph-based location identification algorithms that make use of graph matching. Similarly, Yin et al. [143] present a parallel semantic feature encoding module that uses a divergent place learning network to increase the reliability of place identification by extracting the various kinds of semantics (tree, building, etc.). Recently, Paolicelli et al. [97] integrated visual appearance and semantic context using a multi-scale attention module for stable feature embedding. In recent years, new loss measures have also been a big part of making location identification more accurate. As a common measure, triplet loss is used by current place recognition algorithms like [129, 8] to build query-positive-negative pairings by (based on the Euclidean distance). In this approach, favorable references are downplayed while negative ones are amplified from the beginning of the inquiry through its conclusion. When training, Yin et al. [141] offer a rotation triplet loss based on the triplet loss, which helps enhance the model's performance regardless of its orientation. In addition to triplet loss, angular loss is often used as a learning location descriptor. Furthermore, based on the aforementioned characteristic, Yin et al. [143] established a divergence loss measure that may mandate the place feature learning technique for distinct semantic structures (e.g., trees, buildings, roads, etc.). As mentioned above, Li et al. [71] approach place identification as a classification issue, ignoring the aforementioned loss measures. Even when all of these methods are taken into account, it is still impossible to make a system that can recognize places in the real world, where there are an infinite number of ways that the environment and situations can change. Lifelong education based on familiar landmarks might be the answer. Learning throughout one's life (or "continuous learning") is meant to keep one's skills and knowledge current in the face of constant disruption. Keeping the required candidate images inside the memory zones preserves the Hidden Markov Model's lifetime learning property [29]. The graph-pruning-based approach for lifetime LiDAR SLAM provided by Kurz et al. [69] may reduce vertices and edges to keep the graph size manageable while repeatedly visiting the same places.

2.5.4 Efficiency and Robustness

Robustness and efficiency are two qualities crucial to the success of long-term localization in practical settings. The simplest location matching method employs a frame-based re-localization strategy [87, 104], which may offer effective localization for stationary and temporary navigation tasks. In contrast, real-world applications are rife with sudden shifts, gradual alterations in appearance over time, and recurring visual cues. The aforementioned constraints limit the utility of place recognition methods that rely on a single frame. Sequences that match [85] are preferable to single images because they increase robustness to small-scale variations in the scene and decrease the number of false positives caused by similarities between the images themselves. The sequence-to-sequence (S2S) matching technique used by SeqSLAM [85] and similar publications [101, 123] is a brute-force approach. New learning-based algorithms [76, 140, 142, 19] have improved location identification performance in long-term localization tasks, building on the foundation laid by SeqSLAM. Matching accuracy is improving, but the process is time-consuming and not suitable for real-time applications. Particle filters, the estimated world's closest neighbors, and the Hidden Markov Model are only a few of the strategies presented in [75, 114, 48] to boost SeqSLAM's performance. It is possible that a huge number of reference sequences might reduce the effectiveness of the methods. The use of dynamic query sequences and binary descriptors by [126] and [9] enhances SeqSLAM. The effectiveness of these techniques is extremely context-dependent, making them vulnerable to failure in dynamic, difficult settings. may strike a happy medium between matching precision and efficiency, benefiting long-term navigation by way of improved global localization accuracy. Bampis et al. [10] propose a sequence bag-of-words with a unique temporal consistency filter that can gain from sequence matching while still maintaining real-time performance on a tablet, as opposed to the burst-force searching through the difference matrix used in SeqSLAM. A combination of a compact and sparse neural network (FlyNet) with a continuous attractor neural network (CANN) to capture the sequence of observations has been shown to outperform SeqSLAM [19]. Similar to [19], Garg et al. [41] also use a temporal convolutional network to produce a sequence global descriptor, which may be used to build a hybrid location recognition system by combining global matching with the local sequence matching.

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

Chapters 5 and 6 of our dissertation are inspired by Knopp et al. [65], which improves the database by excluding traits that are a good match only for locations far away. However, this method is not optimal for extremely big databases due to the refining cost, which grows quadratically with the size of the database. Alternatively, some approaches generate distinct [46, 17, 136, 21] classifiers for each geographic location, resulting in naturally weighted characteristics for each region. However, a model has to be trained for each conceivable location in the dataset if you use these methods. Our techniques are likewise examples of data-driven approaches. In contrast to previous approaches, ours uses a robust representation mechanism to separate important from irrelevant items in response to a single query.

2.7 Visual Place Recognition Using Feature Selection

The place recognition process might be muddled by the use of obfuscating signals introduced by features taken from things that are universal across geographies. Windows, fences, and trees all fall under this category since they may be found in a wide range of places and are clearly distinguished from one another. Attempts have been made to use geotags in the database to choose attributes that are discriminatory with respect to location. Schindler et al. [111] to created a word tree with information specific to their geographic area. Doersch et al. [30] look for geographic patches that are both common in a given location and distinct from those in other regions, rather than trying to identify characteristics that are particular to a single landmark. Even the post-processing phase of geometric verification used by most retrieval methods to whittle down the candidate set may be fooled by pervasive components. It is possible that the number of inlier matches between a query and a non-related picture is greater than that between the query and its related images due to ubiquitous components. Generic structures with recurring patterns are often to blame. Addressing this issue, Sattler et al. [109] offer a feature weighting approach that compares photographs on the shortlist with their accompanying GPS-tags to discover and down-weight such geometric bursts.

Using a Global Positioning System (GPS) tag associated with a training query, [63] selects a collection of positive examples based on photos that are physically located within 50 meters of the tag. Geometric verification is used to further narrow down the positive choices, given that photos with the same GPS coordinates may be captured by just angling the camera in a different direction. In order to provide counterexamples, the authors simulate the picture geo-localization procedure inside the training batch, selecting as a negative candidate the best image that was received at least 225 meters distant from the GPS position of the training query at each iteration. The layout of PBVLAD-based predictor features is shown in the referenced figure 2.9.

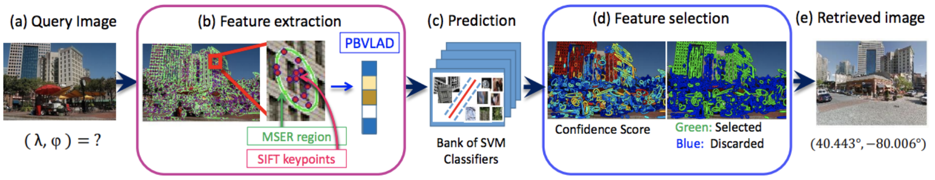


Figure 2.9: PBVLAD approach [63]. Given an input query image with an unknown geo-location (a), MSER regions and SIFT keypoints form bundled features, which are then represented by PBVLADs (b). Features go through a pre-trained bank of SVMs that outputs binary predictions about a feature being "good" 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).

Our methods (Chapters 5 and 6) are explicitly trained to discriminate geographically ubiquitous visual elements; they automatically discover them and adjust their weights in our image representation.

2.8 Discussion

In this thesis, our goal is to propose efficient visual place recognition methods. We apply the same logic to works whose distinguishing features are [63, 8, 55, 43]. Prior to selecting features, we must first identify robust image representation methods for determining which features are relevant or not for an input image.

2.8.1 Mini-Batch VLAD for Visual Place Retrieval

To begin, we address the issue of computational expense caused by the high dimensionality of visual features used in representation for a visual place retrieval task. There have been a lot of studies written on how to get better retrieval results. When it comes to encoding images in a retrieval system, Jegou et al. [57] suggest a method known as the Vector of Locally Aggregated Descriptor (VLAD). VLAD is based on a vector representation of an image that aggregates SIFT [77] descriptors using a locality criteria in the feature space. To fix the inefficiency of the VLAD retrieval system, the team developed HVLAD [32]. Growing the vocabulary from hundreds to hundreds of thousands of words may improve search accuracy, but at the expense of a significant increase in computation cost when using flat quantization. To strike a balance between the discriminability of the descriptors and the computational complexity of the model, they suggest a hierarchical multi-VLAD. They built a tree-structured hierarchical quantization (TSHQ) to speed up the VLAD calculation while dealing with a large vocabulary. An approach for indexing the SIFT descriptors of discovered SIFT [146] interest points in reference images is proposed by Zamire and Shah. The query image's discovered SIFT descriptors are used to pinpoint the tree's location. Each query descriptor has a vote for where its closest neighbor is located in the query image, allowing for accurate localization. Kim et al. propose a Per Bundle Vector of Locally Aggregated Descriptors (PBVLAD) for feature representation, wherein a vector of locally aggregated descriptors (VLAD) [63] is used to characterize a maximally stable (MSER) [82] region based on multiple scale-invariant features (SIFT) [77] detected in that MSER. Unfortunately, huge datasets with many items, like street view, are infeasible for the currently available approaches, which all concentrate on increasing retrieval accuracy. As a first contribution, we look at using mini batch k-means in VLAD [57] to address the dimensionality issue with visual vectors. We provide a computationally cheap and effective approach for retrieving images.

2.8.2 Aggregated Binary Local Features for An Efficient Visual Place Retrieval System

The second main contribution of this thesis is the suggestion of using aggregated binary local features to solve the issue of efficient image retrieval. The study that comes closest to ours is on aggregating binary local descriptors for image retrieval [38], but it focuses on a different aspect of the problem. In order to boost retrieval performance, the authors of [38] investigate how aggregations of binary local features interact with the CNN pipeline. As part of our contribution, we explore the use of binary features with aggregated representation for visual location retrieval as a means of lowering the cost of feature extraction and representation. Furthermore, very high-dimensional descriptors like the "bag of visual words" [118] and the Vector of Locally Aggregated Descriptors (VLAD) vlad are required to get excellent retrieval accuracy on such datasets. Retrieval challenges often include aggregating VLAD [57] into smaller, more manageable descriptions. As a result of the binarized VLAD's efficiency in computing the Hamming distance [38], we retransmitted the compressed to binary aggregated descriptors. with the goal of improving retrieval accuracy for a massive image database.

2.8.3 Predicting Good Features Using A Hybrid Feature For Visual Place Recognition

Visual geolocalization is a well-researched topic, and many studies have been conducted. The closest research to ours, are the visual geolocalization applications focusing on city-scale methods and using an image-retrieval technique to solve the problem [50, 8, 55, 63]. Basically, the image geolocalization process has two steps. One is the feature extraction and representation step, and the second is computing the similarity between input and image and reference dataset. Many retrieval-based solutions use handcrafted methods for feature extraction steps such as SIFT [77], which are robust against the variability between the input image and the reference dataset. Zamir and Shah [146] constructed image representations using local invariant features [77].However, SIFT is designed to to produce a huge number of descriptors for each data point, making a similarity search computationally expensive. Feature aggregations such as the VLAD [57] are performed to reduce the dimensionality of the descriptor vector and confirm particular features that are most valuable for the visual geolocalization task. Jegou et al. [57] introduced VLAD for an image retrieval system. In VLAD, the final descriptor includes the difference between a feature and the closest visual word to that feature. Moreover, the implied idea behind VLAD representation has inspired many studies [63, 8, 5]. Kim et al. [63] Kim et al. introduced the per-bundle VLAD (PBVLAD) approach. In PBVLAD, local SIFT features are discovered inside MSER. Furthermore, Mini Batch VLAD [5] is an extension of VLAD [57] they applied to use mini-batch k-means clustering between two vectors instead of using k-means clustering. Traditional approaches such as VLAD[57], PBVLAD [63], Mini Batch VLAD [5] and SIFT [146], on the other hand, are unable to capture higher-level structural information.

In comparison to traditional methods, deep-learning-based algorithms have recently demonstrated promising performance in extracting features in computer vision tasks. Deep learning is based on extracting high-level features using multi-layer networks, which are more resistant to changes in appearance. Sunderhauf et al. [**?**] used AlexNet [67] to extract features from the ImageNet dataset. PCANet was proposed by Xia et al [138]. for extracting features as picture descriptors. Sun et al. [121] suggested a CNN-based point-cloud-based place recognition task. Camara et al. [16]introduced a two-stage visual place recognition system that encodes images using the activations of multiple VGGNet layers. SequeezNet [54] is a neural network with fewer parameters and a smaller size. Even if those methods are excellent, the networks utilized were built for picture classification tasks and are not ideal for image geolocation tasks.

The models employed to extract these traits aren't built to deal with the drastic changes in the environment that are prevalent in visual geolocation assignments. Deep-learning-based approaches such as NetVLAD [8] and PatchNetVLAD [50] have demonstrated promising performance in visual localization tasks. NetVLAD [8] is a VLAD layer that is added to the end-to-end architecture. The convolutional neural network (CNN) is cropped in the last layer, and we create a new layer based on VLAD [8]. PatchNetVLAD [50] is a recently proposed visual place recognition algorithm. PatchNetVLAD [50] obtains patch features from the NetVLAD residuals and uses them to exploit the local and global descriptors. Features are matched in Patch-NetVLAD utilizing joint closest neighbors and geometric verification using RANSAC. However, those methods are computationally expensive, and when the number of features is lowered, consistent retrieval performance cannot be maintained.

To capitalise on both handcrafted and deep-learning methods, some works have combined CNN descriptors with local detectors. In that event, as in the VLAD method, features extracted are gathered into a single descriptor. However, all convolutional and fully connected representations are computationally expensive by requiring the large number of parameters. In our current study, conversely, we concatenate the features after fusing the deep and handcrafted features to overcome the limitation of computational cost in the previous representations. We solve the problem by extract multiple patches using pretrained deep learning representation combine with hand crafted representation. To this end, we combined different methods of handcraft [77, 57, 63, 5] with two deep learning methods, AlexNet [67] and SqueezeNet [54] for the feature extraction and representation phases. Also, one advantage of our hybrid representation is effectiveness when used with classifier, as we cast the geolocation problem as a classification task.

However, not all image content is useful or relevant for geolocalization, and it requires the elimination of misleading information. To address this problem, researchers have attempted tofeature selection as a classification problem [63]. Other works have tried to dismiss features appearing in multiple geographical places [61]. There has also been work on feature selection for the tasks of image retrieval. Conversely, to predict a useful feature for geolocalization on a city scale, Kim et al. [63] have trained a bank of SVMs. Kim et al. [63] employ geotagged images from social media platforms to generate their own training data. Our work is closely related to that of Kim et al. [63], but with different focuses. Similarly, Kim et al. [63] attempt to predict useful features to find.the correct geolocalization, leading to improved performance in terms of robustness, we show in our Experiment with making geolocalization performance more efficient in terms of searching time.robustness and accuracy while reducing a range of features.

2.8.4 Efficient Visual Place Recognition System by Predicting Unique Features

Although geolocation is used to select related street-view images, there is no visual match ground truth in this sample.Positive and negative samples of images captured with a geotag reader are used for training. NetVLAD [8] uses a learnable layer based on VLAD [57] to aggregate local descriptors and allocate them to cluster centers. Using context-aware feature reweighting and effective hard negative mining, CRN [55] chooses the best contexts for localization. Utilizing a self-supervised approach based on image-to-region similarities, SFRS [43] uses the NetVLAD infrastructure. The algorithm is able to locate challenging positive images with the use of these commonalities, which create soft labels. A patch-based local descriptor approach, Patch-NetVLAD [50] reorders the top 100 results from NetVLAD. Multiple studies [63, 8, 55] have shown that training with negative examples forced by these approaches results in accurate localisation. However, there is no theoretical basis for the claim that eliminating negative traits would result in subpar localization accuracy.

In addition, the calculation involved in determining if two features are comparable is an issue for all of these approaches. In contrast to previous approaches, we predicted unique features by selecting features from negative cases during an offline phase. We were able to minimize the number of features while maintaining accurate image recognition using this method. The retrieval phase is completed with enough precision and at a reasonable computational cost.

2.9 Visual Place Recognition and Retrieval Datasets

2.9.1 Visual Place Retrieval Datasets

In our research, we focus on geolocating and retrieving images on a challenging outdoor environments, so we consider datasets that cover only one city. Visual place retrieval is evaluated using standard benchmark datasets such as the Oxford5k [1] ,and Paris6k [2] datasets. These benchmark datasets are very small. For instance, the 55 query images of 11 instances each that were included in the first releases of the Oxford5k and Paris6k datasets in 2007 and 2008 respectively are still frequently utilized today. Both datasets only include images from one city, therefore their findings may not apply to other areas [135]. While there are other single-city imagery datasets, such as Rome 16k[3], San Francisco Landmarks [4]; 24/7 Tokyo(not avalible online) and Paris500k [5], these datasets are not freely available for public use. The datasets mentioned thus far are summarised in Table 2.2.

[1] https://www.robots.ox.ac.uk/ vgg/data/oxbuildings/

[2] https://www.robots.ox.ac.uk/ vgg/data/parisbuildings/

[3] https://sites.google.com/site/greeneyesprojectpolimi/downloads/datasets/rome-landmark-dataset

[4] https://exhibits.stanford.edu/data/catalog/vn158kj2087

[5] https://www.vision.rwth-aachen.de/page/paris500k

Table 2.2: Geotagged, city-scale datasets for visual place-retrieval [135].

Dataset name	Year	Landmarks	Test	Index	Annotation
Oxford	2007	11	55	5k	manual
Praise	2008	11	55	6k	manual
Rome 16k	2010	69	1k	15k	geotag+sfm
San Francisco	2011	-	80	1.7m	street view
24/7 Tokyo	2015	125	315	1k	smartphone
Paris500k	2015	13k	3k	501k	manual

2.9.2 Visual Place Recognition Datasets

Season, perspective, and lighting all play a role in the variety of datasets offered by the academic community in the field of visual place recognition. In general, there aren't a lot of databases that can be used for accurate visual geo-localization. In this section, we identify the datasets that will be useful for evaluating and contrasting different kinds of visual geo-localization methods. There are several datasets for city-scale surroundings that may be downloaded from websites like Flickr, Panoramio, and Google Street View.

Google Street View Dataset

The Google Street View Dataset [6] introduced by Zamir and Shah [147] includes 102k images that were collected in a completely automated way from the Google Street View website, mostly in the cities of Pittsburgh, Pennsylvania, and Orlando, Florida. There are complete 360-degree panoramas in this collection, with around 12 meters separating adjacent spots. Each placemark in the database is represented by five images: four from the side and one from above. This dataset simply includes the image and its coordinates; no other

[6] http://crcv.ucf.edu/projects/GMCP$_{Geolocalization}$/

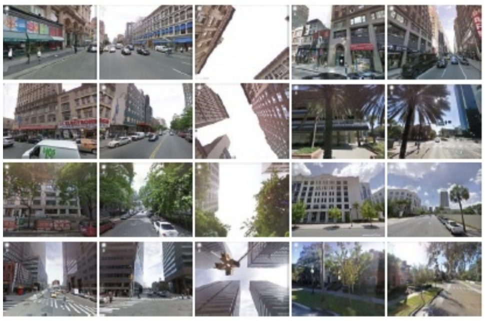


Figure 2.10: An example of panorama image from Google Street View [147].

information, such as the ground truth, is provided.

IM2GPStestsets

Six million geo-tagged images were downloaded from the Flickr site and used to train the IM2GPS [51] method developed by Hays and Efros citeim2gps. The test sets, which each consist of a few hundred images, may be seen in their entirety on the website. [7]

Pittsburgh 250k

In the field of visual geo-localization, the Pittsburgh 250k dataset [8] is often used as a standard. It has 250k Google Street View screenshots that were downloaded to it. 24k requests for images from Street View at various times of day and years. You must make a special request to access this dataset. Pittsburgh 250K, comprised of 254,064 images from GSV, was utilized to create the geotagged reference dataset. Images were shot in 10,586 distinct spots, mostly in Downtown Pittsburgh, with a 640x480 pixel resolution. Images were shot from two different heights and angles at each location: floor level (0 degrees) and 30 degrees above the ground.

[7] http://graphics.cs.cmu.edu/projects/im2gps/

Tokyo 24

The Tokyo 24 [125] dataset presents significant difficulties. Daytime Google Street View images make up the bulk of the 76k included in the collection. The 315 images included in the query were all shot with either low or very high lighting. Further, access to this data collection is restricted and must be requested.

2.9.3 Geo-tagged Image Collection

As we see in the preview section, there is a paucity of available data sources for the task of visual geo-location. Datasets tend to cover more than just the one city, and some of them are not freely available, or do not include GPS tags. Our work in this thesis requires a geotagged dataset of images covering the same geolocation as the reference dataset [147]. We therefore propose to collect own geotagged images from social media platforms. The output can be divided into two categories: 1) Flickr Images, and 2) Metadata. The metadata includes information such as photo IDs, titles, bookmarks, comparisons, designs, and the photographer. Textual information (title, symbol, description) is provided by the user, but location data is provided in two ways: 1) by in-camera GPS receiver, 2) by internet services.

Our geotagged dataset contains 720 images that can be used for training, and 100 images for testing. The images can be used for many computer vision tasks such as place retrieval, visual place recognition, and image classification. The images include street-level images taken from multiple perspectives, at different but nearby locations. Our collected images cover cityscape views, major buildings, and outdoors landmarks in Pittsburgh, PA. The Flicker API allows im-

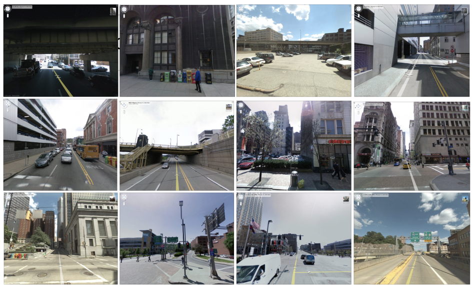


Figure 2.11: A set of images from our collection of geotagged images from Flicker API

.

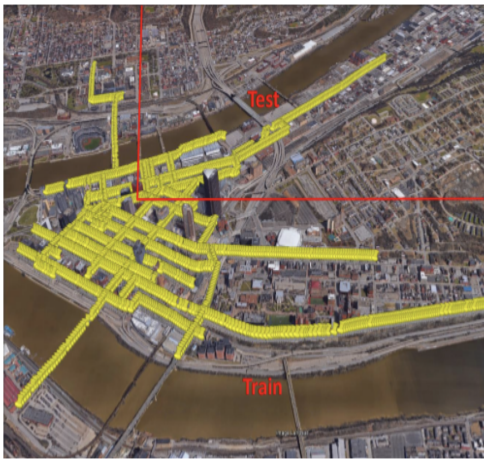


Figure 2.12: Distribution of the area in Pittsburgh city, PA that covered the same area in Google Street View Dataset. [147]

ages to be selected by tag. For example, we can search by objects, places and buildings such as lamp-post, pizza, church, etc. For our dataset, we searched by GPS tag, using geo-location (latitude and longitude) from the Google Street View Dataset provided by Zamir and Shah [147]. Figure shows the distribution of the covered area in Pittsburgh, PA from [147], that we used the same GPS tags for searching about images.

The search returned many duplicate images, which we removed. Finally, we manually reviewed the obtained images in order to remove all images that did not meet the content criteria, e.g. indoors shots, lacking buildings, too abstract, or otherwise inappropriate for describing the actual place. The resulting dataset comprises relevant images with approximate location tags in their filenames. Note that images used for testing have manually verified GPS-tags. For more details about collecting images and implementation details, see the Appendix.

2.10 Evaluation methodology

2.10.1 Evaluation metric for visual retrieval

Average Precision (for each query) and Mean Average Precision (for all query) are the most widely used in content baaed image retrieval field. If the retrieval method as a solution of recognition task, ROC curve and AUC are applioed as the measure. They also reflect both precision and recall information. In our research, we follow the same evaluation method in [63, 8]. They used Mean Average Precision and ROC curve to evaluate the accuracy for retrieval method.

2.10.2 Evaluation metric for visual place recognition

For our assessments, we use the standard evaluation as [63, 8, 55, 43, 50]. If at least one of the best N images returned by the search is within d =25 meters of the query, then the localization is considered accurate. The proportion of properly localized queries over all values of N is called the recall@ N.

Chapter 3

Mini-Batch VLAD for Visual Place Retrieval

This chapter includes the first contribution to this thesis. In terms of improving the efficiency of a visual place recognition system, this chapter investigates the visual place retrieval of an image query using a geotagged image dataset. Then, we compare and visualize the results of our proposed method (Section 3.3). We also compare popular approaches proposed in the literature that tackle the visual place recognition problem using traditional extraction solutions (Section 3.4). Most of this chapter was taken from our paper, "Mini-Batch VLAD for Visual Place Retrieval" [5].

3.1 Introduction

3.1.1 Motivation

The availability of images with their geolocations, coupled with additional information (e.g., text, time stamp,etc.), has led to many applications such as object geo-localization [68] and flood monitoring [1]. However, many images lack accurate (or any) GPS information; for instance, tweets often include GPS information about where people tweet, but pictures posted in a tweet may have no GPS tag. In order to recover lost GPS information, Bulbul et al. proposed to query the Google Street View image database [13].

In general, the pipeline for recognizing a certain place using a single visual query has three successive steps. This pipeline is applicable to large datasets, such as city-scale First, regions are located in the query image. Second, descriptors are generated over these selected regions in order to provide an accurate representation of the query image. Finally, this representation is matched against geotagged images in the reference dataset, and the GPS information of the retrieved reference is obtained for the query image (see chapter 2)).

Local Scale Invariant Feature Transform (SIFT) [77] has been used to describe interesting parts of images with powerful features. SIFT is also robust to photometric and geometric changes [72, 146]. Therefore, SIFT has an important role in image retrieval. The Vector of Locally Aggregated Descriptors (VLAD) [57] have been shown to be powerful local features for image geo-localization and retrieval. VLAD represents an image as a single fixed-size vector using K-means clustering. The issue, however, with VLAD is the dimensionality of visual features and the computational load of sample distances in a large image dataset.

3.1.2 Contribution

In this chapter, we propose instead to learn VLAD by using mini batch k-means clustering [112] (MBVLAD). One of the local features that can be used for image place retrieval and recognition is the Vector of Locally Aggregated Descriptors (VLAD). VLAD describes an image by comparing its local feature descriptors to a previously computed codebook. Generally, a visual codebook is generated from k-means clustering of the descriptors. Among the works aimed at improving local feature descriptors for visual place recognition, Kim et al. [63] propose PBVLAD as a novel method to locally integrate SIFT features detected with a Maximally Stable Region (MSER) blob [82]. Per Bundle Vector Agregated Locally Vector (PBVLAD) is the name of the descriptor. The purpose behind this descriptor is to find a robust local feature descriptor against geometric and photometric changes. Each MSER is described by VLAD based on multiple features detected in the region. As an accumulation of variance between the descriptors that are allocated to the visual word and the centroid, a sub-vector of the PBVLAD was proposed. In fact, this processing takes place on each image patch, which is time-consuming when computing VLAD descriptors for large datasets.

In this chapter, our goal is to design an accurate image retrieval method with affordable computation expenses. We focus on the dimensionality of visual features, which is not trivial, and the computational load of sample distances in a large image dataset, which is challenging. We propose to use mini-batch k-means clustering to compute VLAD descriptors (MBVLAD). In particular, we try to improve local feature descriptors for visual place retrieval by using an alternative clustering algorithm. We adopt the mini batch k-means algorithm that was proposed by Sculley [112], to compute VLAD [57]. The mini batch k-means algorithm outperforms the original k-means algorithm on large-scale datasets by lowering computation costs and data processing time.

3.2 Design

3.2.1 Mini Batch Vector of Locally Aggregated Descriptors (MBVLAD)

In this section, we will discuss how to use mini-batch k-means clustering in feature-based visual place retrieval. The pipeline of feature-based visual place recognition entails three stages: First, select a single query image with an unknown location as an input. Second, extracting features using SIFT keypoints. The final stage is to compute a vector representation using the k centroids to match the database. Figure 3.1 depicts the proposed pipeline used in this paper. The following sections discuss each pipeline stage in detail.

3.2.2 Mini-batch K-means

Our goal is to retrieve images using parts of an input image for geo-localization. One challenge of feature learning is the scale of the dataset. In the case of learning from a city dataset, the computational load is very high. Here we propose a Mini Batch Vector of Locally Aggregated Descriptors (MBVLAD). The key idea is to aggregate features using a vector with a fixed size and to learn the vocabulary words using the mini-batch K-means clustering algorithm. In this way, we reduce computational load, and it is still convenient to use standard distance measures to retrieve relevant images.

The concept of a mini batch k-means algorithm was proposed by Sculley [112], in two iterative steps. The first step is to form a mini-batch by taking samples randomly from the dataset, and then each sample in the mini-batch is assigned to its nearest centroid. In the second step, the centroids are updated as the mean over their associated samples in the mini batch. These two steps are then repeated for several iterations. It should be kept in mind that the number of iterations used in the mini batch k-means algorithm affects clustering quality: the more iterations used in the mini batch k-means algorithm, the better the clustering result quality. See the algorithm for more details 1.

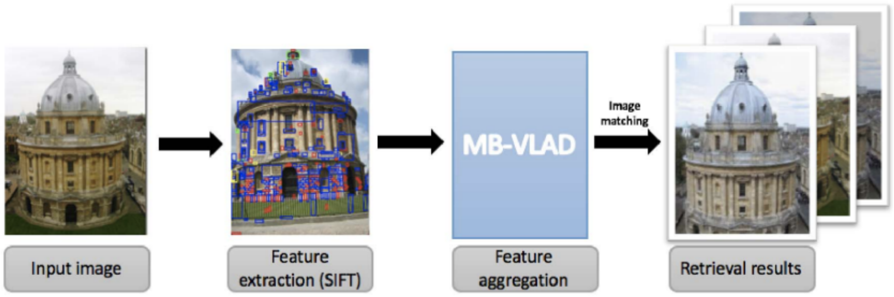


Figure 3.1: Our proposed pipeline to process a input (query) image.

Algorithm 1 Mini batch k-means clustering.

```
Given $k$, mini-batch size $m$,iterations $t$, dataset $X$
Initialize each $c \in C$ with an x picked randomly from X $v \leftarrow 0$
for $i = 1 \rightarrow t$ do
    $S \leftarrow b$
    for $x \leftarrow S$ do
        $d[x] \rightarrow f(C, x)$ //Cache the center nearest to x
    end
    for $x \in M$ do
        $d[x] \leftarrow c$ // Get center for this x temporary
        $v[c] \leftarrow v[c] + 1$ // Update per-center counts
        $\eta \leftarrow \frac{1}{v[c]}$ // Get per-center learning rate
        $c \leftarrow (1 - \eta)c + \eta x$ // Take gradient step
    end
end
```

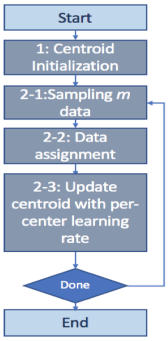

Start
1: Centroid Initialization
2-1:Sampling *m* data
2-2: Data assignment
2-3: Update centroid with per-center learning rate
Done
End

Figure 3.2: Mini K-means Clustering.

3.2.3 Vector of Locally Aggregated Descriptors (VALD)

The original VLAD approach [57] builds a codebook dictionary $C = \{c_1, c_2, \ldots, c_k\}$ from $m \geqslant k$ feature vectors in the reference dataset. To generate the dictionary, a k-means clustering algorithm is used. For an image having m descriptors $I = \{x_1, \cdots, x_m\}$, the VLAD coefficient V_i is computed by accumulation over these descriptors in cluster c_i:

$$V_i = \sum_{x \in I / q(x) = c_i} x - c_i \tag{3.1}$$

where $q(x)$ is the cluster associated with x.

The final VLAD representation is a concatenation $v = \{v_1, \ldots, v_i, \ldots, v_k\}$ followed by L_2 normalization $v : v / \|v\| \rightarrow \tilde{v}$. Thereafter, VLAD encodes features by computing the residuals [7], and the residuals are stacked together as vector v. In this study, we propose to replace the set of centroids inferred by k-means c_i algorithm [112] with cm_i the cluster centers from mini batch k-means. When generating the dictionary, a mini batch k-means algorithm is applied. Clustering input data are unnormalized SIFT descriptors before adding them into mini batches. When the dictionary is generated, two normalizations are applied to compute the final VLAD. First, the power law normalization V_u is applied: for $u = \{1, \ldots, Kd\}$, $V_u = sign(V_u)|V_u|^{\alpha}$ [57]. Then L_2 normalization is used.

The similarity grouping is performed by distance measurement. We apply Euclidean distance to compute the similarity metric. When searching for the closest VLAD vector, the one with the lowest Euclidean distance is selected.

The dimension of MBVLAD can be reduced when searching for the nearest neighbor by using Principal Component Analysis (PCA). We fit PCA in offline mode and then transform all MBVLAD feature vectors in the dataset. PCA is performed on subvectors v_i derived from each visual word c_i. We generate a coarse vocabulary of 128 visual words (16,384-dimensional MBVLADs in raw form). Thereafter, some major components are used to reduce the dimensionality of VLAD vectors using 128 visual words.

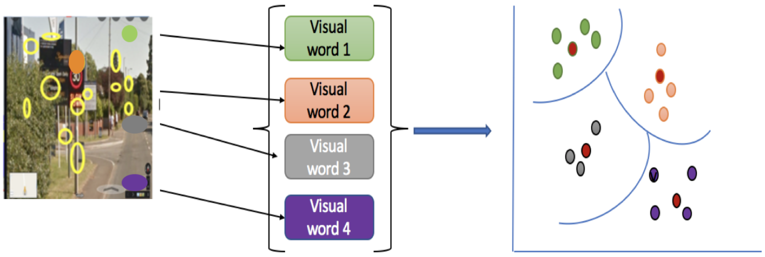


Figure 3.3: VLAD method.

3.3 Experimental Assessment

3.3.1 implementation deatails

We implement our method using Python 5.3. Local features that were extracted by using OpenCV (Open Source Computer Vision Library) [12]. In our experiment, we set the maximum number of iterations over the complete dataset to 100 and the size of the mini batches m

to 500,000. Typically, the VLAD vector size is kxd-dimensional. In our experiment, k ranges from 16 to 128 and d is 128.

3.3.2 implementation detset

For standalone MBVLAD descriptor evaluation, the Oxford building dataset [102] was used. It is usually called the Oxford 5K. It consists of several image subsets, assembled together by an image quality measure (mainly the percentage of represented object visibility), with each image set partition labeled as $\{good, ok, ugly, bad\}$. Additional images placed in *query* subset served as the query test set. However, we used the entire dataset to compute the visual dictionary to get the best possible distribution for the calculation of the cluster centroids while clustering.

3.3.3 Experimental results

Performance is evaluated using the mean average precision (mAP), which is the mean of the average precision scores for each query. We evaluate the proposed descriptor MBVLAD on Oxford building datasets (Sec. 2.9) and obtain good mAPs compared to the state of the art for uncompressed descriptors (Sec. 3.3.3), compressed descriptors (Sec. 3.3.3). The robustness of our approach to the choice of centroids is evaluated in paragraph 3.3.3.

On Uncompressed VLAD

Table 3.1 shows the image retrieval performance of uncompressed VLAD [57] and PBVLAD [63], HVLAD [32] and MBVLAD (our). VLAD vectors are typically k by d-dimensional.We use d = 128 in all experiments. The adapted method has a significant improvement over other methods in terms of improving local feature descriptors. Comparing with PBVLAD, our method increases the retrieval performance for image geo-localization by 11%. Figure 3.6 shows examples of our successful retrieval images with high accuracy (true-retrievals / total-images).

Table 3.1: Comparison of the mean Average Precision (mAP) performance of several (uncompressed) VLAD signatures evaluated on the Oxford dataset. Note that, by applying VLAD [57], the result is changed every time when computing the codebook. All the results in this table show the highest result achieved.

Descriptor	# Vocabulary	mAP
VLAD[57]	128	0.33
PBVLAD[63]	128	0.36
HVLAD[32]	128	0.40
MBVLAD(Our)	128	0.47

Figure 3.4 shows the mAP in graphical form. The greater the area under the curve, the higher the reported mAP metric. The curve is generated by applying step-by-step thresholding to the prediction scores. The overall mAP score is 0.47 when we calculate the precision-recall curve. The mini-batch algorithm is more robust to the noise introduced by the random selection of initial centroids, and retrieval performance is not affected.

On PCA compressed VLAD

Table 3.2 shows the retrieval performance of MBVLAD on the Oxford5k dataset, before and after the dimensionality reduction using PCA (k= 128, 64, 32, 16, 8). MBVLAD achieves 0.47, which outperforms other feature selection approaches in the literature.

Table 3.2: Retrieval performance of (our MBVLAD) on the Oxford 5k dataset [102], before and after the dimensionality reduction using PCA (128 vocabulary size). The performance is measured by the mean Average Precision (mAP). Note that we only compare our results with the PBVLAD method because they computed their results after the dimensionality reduction in their work [63].

Methods	Full(16384)	(8192)	(4096)	(2048)	(1024)
PBVLAD [63]	0.36	0.36	0.33	0.26	0.21
MBVLAD (ours)	0.47	0.44	0.40	0.43	0.39

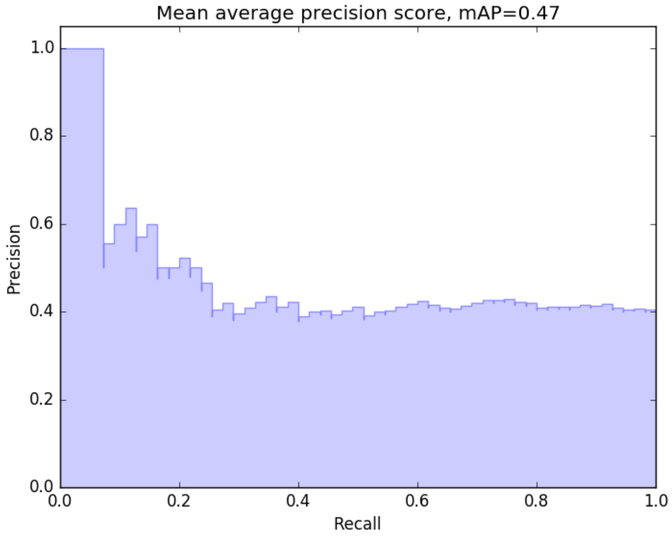


Figure 3.4: The curve shows the mean average precision for our proposed method (MBVLAD).

Sensitivity to initial centroids

When the visual dictionary is generated again, the initial centroids from the mini batch k-means algorithm can be different every time. Table 3.3 shows the mAP for five runs, leading to an average mAP of 0.444 with a standard deviation of 0.0152. Significant improvement is observed in comparison with the state of the art (see Tab. 3.1 for comparison).

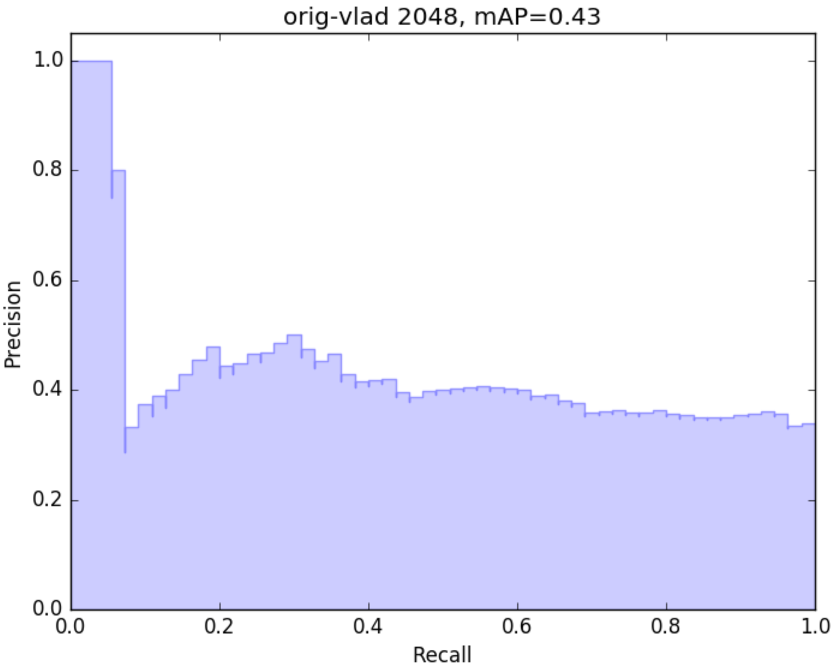


Figure 3.5: The curve shows the mean average precision for VLAD [57].

Table 3.3: Retrieval performance of (our) after generating the visual dictionary for five times.

Run Times	# Vocabulary	D	mAP
1^{st}	128	16384	0.47
2^{nd}	128	16384	0.44
3^{rd}	128	16384	0.43
4^{th}	128	16384	0.44
5^{th}	128	16384	0.44

3.4 Conclusion

In this chapter, we address the problem of finding visual places in a city area using a query image. We propose a mini-batch VLAD descriptor with the goal of improving the performance of a visual place retrieval system under the challenges of geometric changes. Compared with the original K-means clustering approach, MBVLAD has a significant improvement in image retrieval accuracy. The fact that

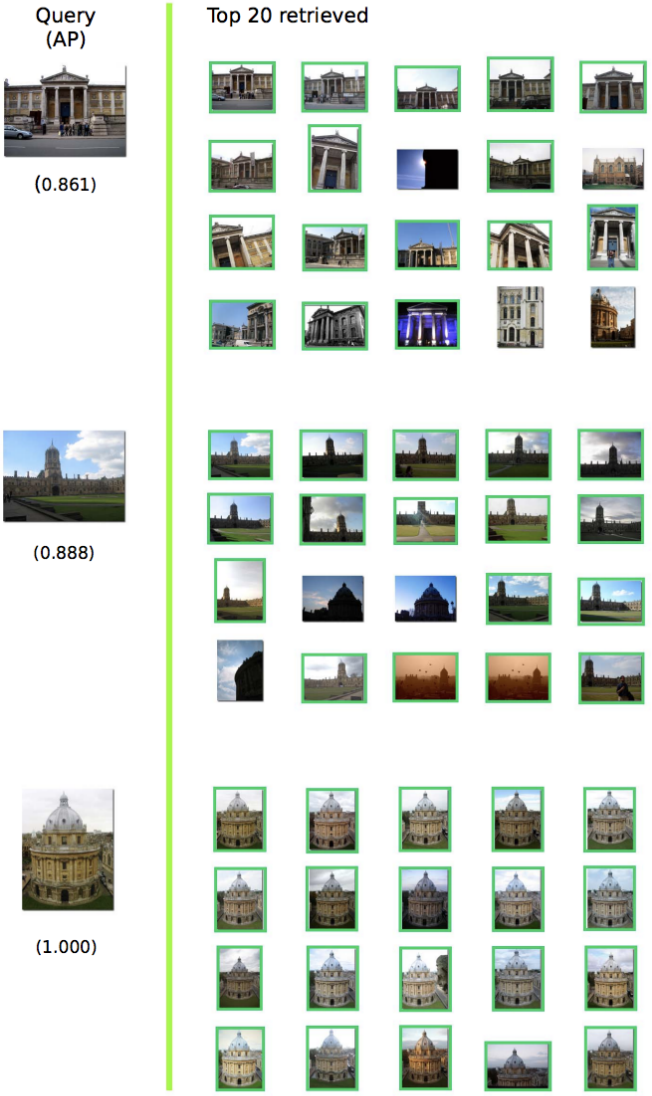


Figure 3.6: Example results (uncompressed MBVLAD): Query images (left) with different sizes, (right) Top 20 retrieved images using our proposed MBVLAD [5].

clustering output is not deterministic and is influenced by initial centroids is a major challenge for the k-means algorithm. From experiments, we find that the mini-batch version is more robust to the randomness of initial cluster centroids as well as significantly reducing computational load.

Chapter 4

Aggregated Binary Local Features for An Efficient Visual Place Retrieval System

In the second contribution, we apply an aggregated binary descriptor (Oriented FAST and Rotated BRIEF (ORB)) with a Mini Batch Vector of Locally Aggregated Descriptor [2] for visual place retrieval. Most of this chapter is published at IMVIP 2020 [2].

4.1 introduction

4.1.1 Motivation

The majority of currently available retrieval systems (like SURF [11] and SIFT [77]) depend on local visual characteristics and approximate closest neighbor search techniques to locate relevant results. Descriptors of local visual features are compared or searched for between individual images. Due to the inefficiency of comparing each descriptor separately, these approaches are best used with very small datasets. In order to cut down on the expense of extracting and matching local visual descriptors, researchers have begun using binary local descriptors like (BRISK and ORB) [38]. The extraction of binary descriptors is substantially faster and more compact than that of non-binary ones [38].

Vector of Locally Aggregated Descriptors (VLAD) [57] is a widely-used technique for aggregating local features. Each visual word's residual distance from its cluster center is calculated using VLAD. Although VLAD [57] is characterized by non-binary features like SIFT [77], the cost of extracting and aggregating local descriptors remains significant. VLAD is one of the most time-tested approaches to location detection and image retrieval. The descriptor, a relatively one-dimensional vector, is meant to characterize the whole image. The extraction of D-dimensional descriptors from affine-invariant detections leads to the construction of a visual vocabulary, which is then clustered into k centers. When searching an image with n local descriptors, the residual from each descriptor to each cluster center must be determined. The residual is then aggregated across clusters and converted into kD-dimensional vectors. The original Euclidean distance is used to evaluate all image vectors [57].

To enhance visual place retrieval, in the previous chapter, we formally introduced MBVLAD [5]. The central concept of MB-VLAD is to learn vocabulary words using a mini-batch k-means clustering technique and to aggregate features using a vector of fixed size. This reduces the amount of work needed to find relevant images, and yet it's still easy to utilize already familiar distance metrics. Principal Component Analysis (PCA) is used to minimize the number of parameters.

In this section, we present an approach to improve the efficiency of the retrieval system in locating the desired visual place within a large database.

4.1.2 Contribution

In the previous chapter, we showed that our proposed MBVLAD [5] is considerably computationally expensive with a large and confusing dataset such as street view images. The problem may come from the computational processing during similarity measurement. To overcome this limitation, we propose to aggregate binary local features that apply Hamming distance. By conducting basic binary comparisons between pixels, binary features may be retrieved rapidly. These processes are relatively computationally inexpensive, and it has been demonstrated that binary feature extraction is faster than gradient-based local feature extraction. In this chapter, binary features are combined into a single vector to achieve efficient image representation by using MBVLAD [5]. To accommodate the fact that binary features are composed of sequences of bits, min batch k-means clustering has used Hamming distance in place of the more typical Euclidean distance. The computational load of representing an image is lightened by techniques like feature extraction and quantization. In order to evaluate the computational efficiency of our proposed method, we compared the results with non-binary methods such as MB-VLAD [5] and SIFT [146]. The datasets used are Google Street View datasets [147] and Oxford 5K datasets [102]. Our experimental results measured with search accuracy (mAP) and search time (s) show that our ORB-MBVLAD is significantly faster with good search accuracy (mAP) compared to our MBVLAD and the other state-of-the-art methods.

4.2 Design

4.2.1 Aggregated binary local descriptor ORB-MB-VLAD

In this section, we describe how to aggregate a binary descriptor using mini-batch k-means clustering in feature-based visual place recognition. The feature-based visual place recognition pipeline consists of three stages: First, as an input, a single query image with an unknown location is queried. The second step is to extract local features using Oriented FAST and Rotated BRIEF (ORB) keypoints [107]. To match the database, the final stage is to compute a vector representation using the k centroids [5]. The proposed pipeline used in this chapter is depicted in Figure 4.1. Each pipeline stage is discussed in detail in the sections that follow.

4.2.2 Oriented FAST and Rotated BRIEF (ORB).

Oriented FAST and Rotated BRIEF (ORB) were first presented in 2011 by E.Rublee et al. [107]. ORB [107] is based off of two other popular tools: the FAST keypoint detector and the BRIEF description.

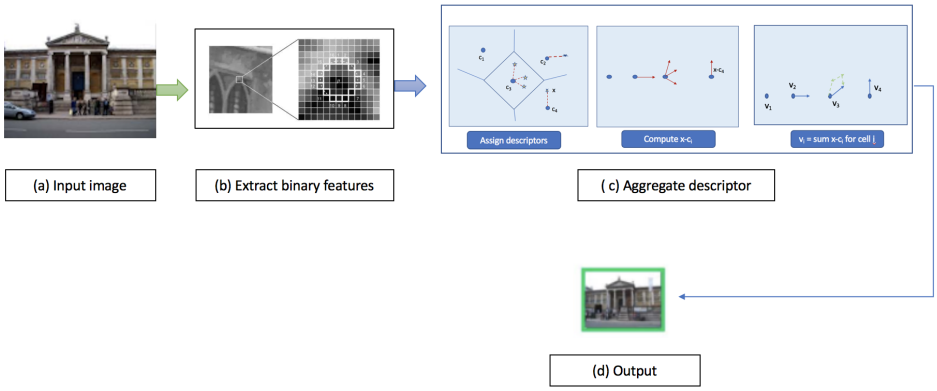

(a) Input image
(b) Extract binary features
Assign descriptors
Compute x-c_i
v_i = sum x-c_i for cell i
(c) Aggregate descriptor
(d) Output

Figure 4.1: Our proposed Aggregated binary local descriptor ORB-MBVLAD method [2]. (a) The query image. (b) Extracting local binary features using Oriented FAST and Rotated BRIEF (ORB) [107]. (c) Aggregated local binary features descriptor using [5]. (d) Retrieved most similar geo-tagged images to (a) as a result.

Both methods are appealing due to their high efficiency and inexpensive price tags. The ORB algorithm starts by finding the position of the key points by FAST and then selecting the N best points. After that, it adds the direction of the points in intensity centroids. Finally, binary descriptors are extracted by BRIEF, and low-correlative pixel blocks are found by the greedy algorithm.

FAST(Features from Accelerated and Segments Test)

For each pixel in an array denoted by i, FAST calculates the average brightness of the 16 pixels in a concentric circle around i. The pixels inside the circle are then sorted into one of three categories (brighter than i, darker than i, or the same as i). In order to determine if a given pixel is a keypoint, we look at the surrounding pixels and choose the ones that are significantly darker or brighter than i if there are more than 8. Therefore, the fast-revealed keypoints inform us where to look for the image's edges to be determined.

Multiscale and orientation components are missing from FAST features. In order to do this, the orb method uses a multiscale image pyramid. A multiscale depiction of a single image, an image pyramid, consists of successive images of progressively lower resolution. The image is downsampled at each level of the pyramid. After the pyramid is constructed, the fast method is utilized to identify important landmarks in the image. ORB is successfully identifying keypoints at each scale, allowing it to pinpoint their locations. In this sense, ORB exhibits some scale invariance. Assign each keypoint an orientation, such as left or right-facing, depending on how the intensity levels change around that keypoint after it has been found. Using the intensity centroid, Orb can detect subtle variations in brightness. The intensity centroid makes the assumption that a corner's intensity is not centered and uses the resulting vector to deduce its direction. To begin, a patch's moments are defined as:

$$m_{iq} = \sum_{x,y} x^i y^q I(x,y) \tag{4.1}$$

We can determine the orientation of corners by using the intensity centroid of an image patch:

$$C = (\frac{m10}{m00}, \frac{m10}{m00}) \tag{4.2}$$

From the center of patch to centroid the angle is given by:

$$atan2(\frac{m10}{m00}, \frac{m10}{m00}) = atan2(m01, m10) \tag{4.3}$$

After calculating the patch's orientation, To get rotation invariance, we may apply a canonical rotation and then calculate the descriptor.

BRIEF(Binary robust independent elementary feature)

In order to represent an object, BRIEF takes all of the keypoints found by the fast approach and transforms them into a binary feature vector. The integers 1 and 0 make up a binary feature vector, sometimes called a binary feature descriptor. Put simply, a feature vector, a string ranging in size from 128 bits to 512 bits, is used to represent each individual keypoint.

To make the descriptor more robust against high-frequency noise, BRIEF first applies a Gaussian kernel to the image for smoothing. Once that's done, choose any two adjacent pixels within a certain distance of that landmark. A pixel's surrounding area is defined by a square patch whose width and height are specified in pixels. The first member of each random pair is selected from a Gaussian distribution with a stranded deviation or spread of sigma centered on the keypoint. The second member of the random pair is drawn at random from a Gaussian distribution with mean zero and standard deviation sigma/2, where sigma is the standard deviation of the distribution. Bits are assigned to 1 if the first pixel is brighter than the second and 0 otherwise. Once again, BRIEF picks a random couple and gives them value. For a 128-bit vector, rapidly repeat this procedure 128 times using the same keypoint. To sum it up, you

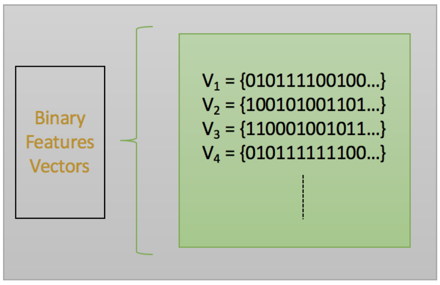

Binary Features Vectors
V_1 = {010111100100...}
V_2 = {100101001101...}
V_3 = {110001001011...}
V_4 = {010111111100...}

Figure 4.2: Binary features vectors in ORB [107].

.

should generate a vector like this for each of the image's landmarks. However, ORB uses RBRIEF since BRIEF is not rotationally invariant (Rotation-aware BRIEF). ORB aims to implement this feature without slowing down BRIEF's performance.

4.2.3 Mini Batch Vector of Locally Aggregated Descriptor (MBVLAD).

For the aggregated phase, we use the extension of VLAD [57]. Vector of Locally Aggregated Descriptor represents an image by a single fixed-size vector using k-means clustering. In [5], we previously proposed learning VLAD by using mini batch k-means clustering to retrieve a place from geotagged dataset. Mini Batch VLAD approach [112] builds a codebook dictionary $C = c1, c2, \dots, ck$ from mk feature vectors in the reference dataset. To generate the dictionary, a mini batch k-means clustering algorithm is used. For an image having m descriptors $I = x1, \dots, xm$, the VLAD coefficient V_i is computed by accumulation over these descriptors in cluster c_i. The combination and L2 normalization that make up the final VLAD representation. After that, VLAD uses the residuals to encode the features. Then, the residuals are stacked together as vector v. In MBVLAD [5], they used the set of centroids inferred by mini batch k-means algorithm with cm_i as the cluster centers. Input data clustering begins with the use of binary descriptors, which are then used to combine data in small batches. When generating the dictionary, we use two normalizations to calculate the final VLAD.

4.3 Experimental Assessment

4.3.1 implementation details

Local features. For binary local features retrieved using OpenCV (Open Source Computer Vision Library), we utilized ORB [107] in the experiments. Up to 2,000 individual local features were discovered in each image.

Visual Vocabulary. In order to construct the VLAD representations, mini-batch k-means clustering techniques [112] were utilized to generate the visual vocabularies. Mini-batch k-means was used on the binary features by first transforming them into real-valued vectors.

VLAD. To do this, we employed MBVLAD [5], our visual information retrieval system, to calculate several encodings of the local feature. Each of these representations has a common parameter, denoted by the number K. It is the same as the total amount of "centroids" (visual words) in an MBVLAD [5]. We also employed principal component analysis (PCA) to lower the dimensionality of the VLAD. We use the same inference from [5] in our experiment, limiting the total number of iterations on the dataset to 100 and increasing the size of the mini batches m to 500000. The usual dimension of a VLAD vector is kd. The value of d=128 was used for this experiment.

4.3.2 Evaluation dataset

We applied ORB with Mini Batch VLAD on the image retrieval benchmark Oxford 5k dataset [102] and Google Street View Dataset[147]. The Average Precision (AP) score is calculated for each of the five searches for a landmark in the Oxford 5K dataset [102]. The mean Average Precision (mAP) score is calculated by averaging these scores (among 55 query photos). related landmark text tags. The connected ground truth answers 55 questions. We ran 5 searches on each of the 11 landmarks. In order to assess the efficiency of feature combinations at scale, we employed a large geotagged dataset in a challenging environment [147].

4.3.3 Experimental results

To quantitatively assess the results, we follow the same evaluation in [146, 63, 38]. We compare the results with the state-of-the-art methods VLAD [57], PBVLAD [63], NetVLAD [8] and MBVLAD [5]. The results are shown in Table 4.1. Note that we repeat our experiment six times because we used k-means clustering, which has the unsupervised nature of clustering. The average values of precision are applied. The performance is evaluated by the mean Average Precision (mAP). The precision defines the relevant image retrieved numbers in response to a query image (number of relevant images retrieved / total number of images retrieved). For query images, we select images from the test dataset randomly. A visual vocabulary of 16 words and 128 vocabulary sizes is applied for all methods in both datasets. All methods are evaluated without dimensionality reduction. For searching time (s), the average response time (s) (retrieval time per query) for each method was calculated. As can be seen in Table 4.2, the suggested aggregated binary descriptor technique yielded faster retrieval performance. Our ORB-MBVLAD averages a retrieval time of 0.076 seconds on the Oxford 5k dataset [102] and 0.521 seconds on the Google Street View image dataset [147].

We use a line chart to graphically represent numerical data as descriptive statistics. The goal of the line charts is to graphically represent the distribution of numerical data, to highlight differences across approaches, and to demonstrate how close our methods are to state-of-the-art algorithms. Line charts show that NetVLAD [8] achieves the best search accuracy for both datasets; however, it is the slowest method time compared with others. Our proposed method came in second in terms of search accuracy, with a mAP 0.49 on the Oxford 5K dataset.

Table 4.1: Several techniques were put to the test on the Oxford 5k dataset [102], and their results were compared in terms of their mean Average Precision (mAP).

Method	mAP	time(s)
VLAD	0.33	0.364
PBVLAD	0.36	1.173
NetVLAD	0.51	1.255
MBVLAD	0.47	0.414
ORB-MBVLAD (Our)	0.44	0.076

Table 4.2: Several techniques were put to the test on the GSV dataset [147],and their results were compared in terms of their mean Average Precision (mAP).

Method	mAP	time(s)
VLAD	0.27	0.944
PBVLAD	0.32	1.384
NetVLAD	0.58	1.171
MBVLAD	0.41	0.722
ORB-MBVLAD (Our)	0.49	0.521

4.4 Conclusion

In this chapter, we offer a technique for efficiently retrieving visual locations by combining binary features. The suggested technique significantly improved computing speed while losing just a little accuracy as compared to state-of-the-art methods. This was because the

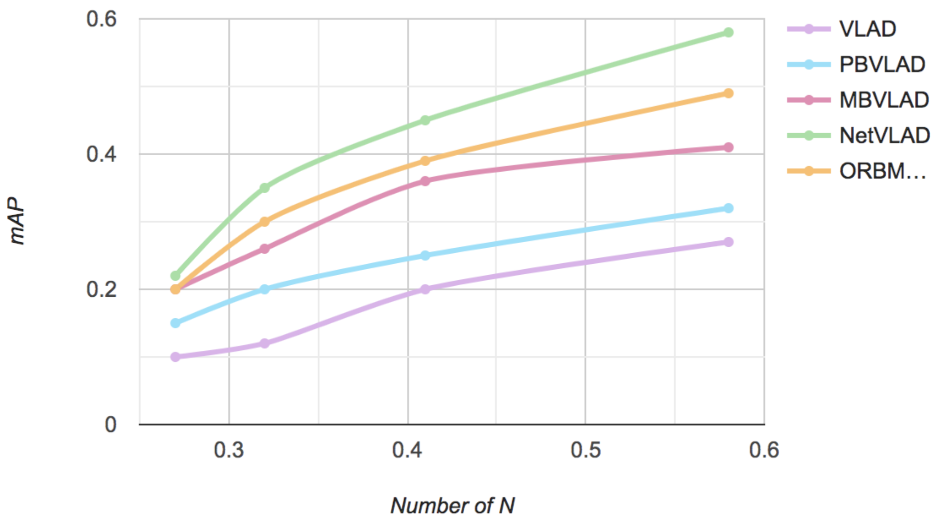


Figure 4.3: Comparison with state-of-the-art. The mAP performance of Ours ORB-MBVLAD compared to VLAD [57], PBVLAD [63], NetVLAD [8] and MBVLAD [5] on Google Street Veiw dataset [147].

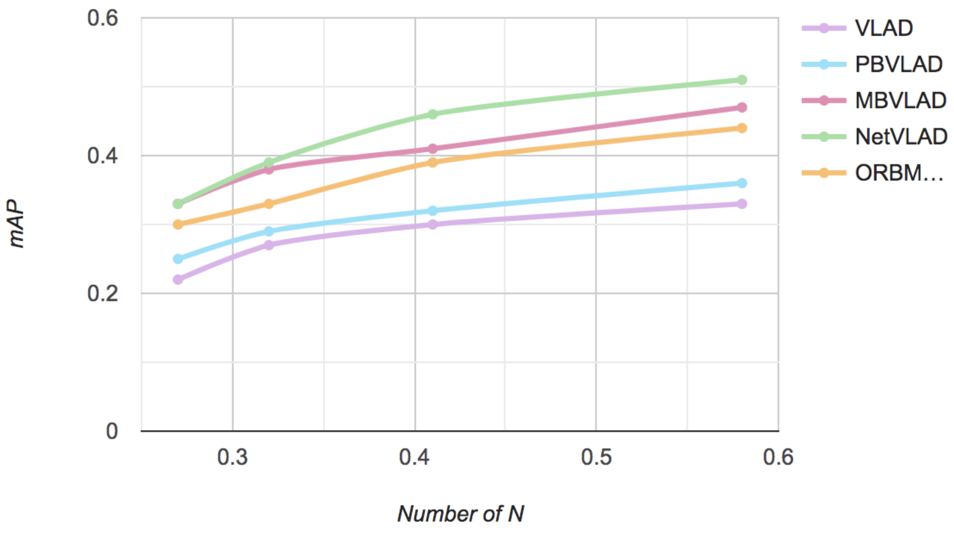


Figure 4.4: Comparison with state-of-the-art. The mAP performance of Ours ORB-MBVLAD compared to VLAD [57], PBVLAD [63], NetVLAD [8] and MBVLAD [5] on Oxford dataset [102].

binary feature itself included a trade-off between accuracy and processing performance. In addition to that, the suggested technique surprisingly outperformed the MBVLAD method, which also used the same representation method, in terms of both accuracy and computing efficiency. Finally, we anticipate that the proposed method will

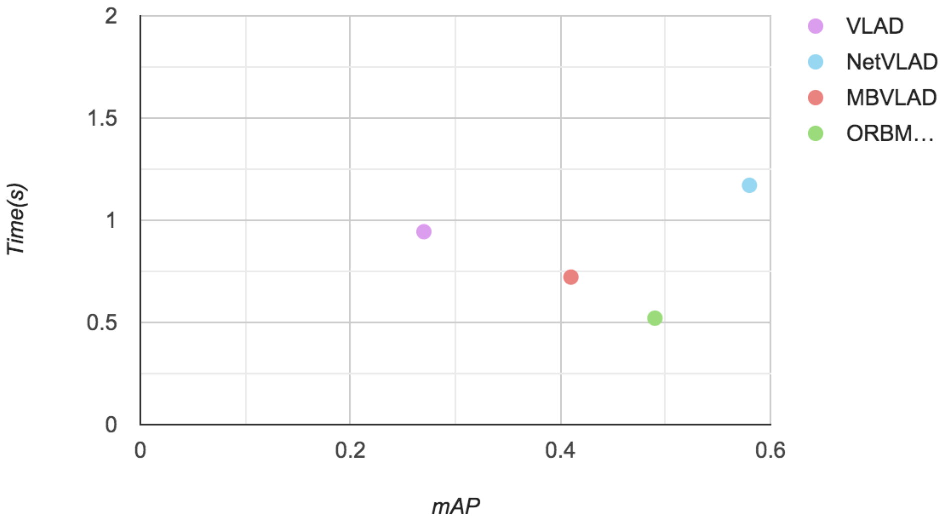


Figure 4.5: Time needed for computation. For the GSV dataset [147], the mAP is shown against the time it takes to analyze a single query image.

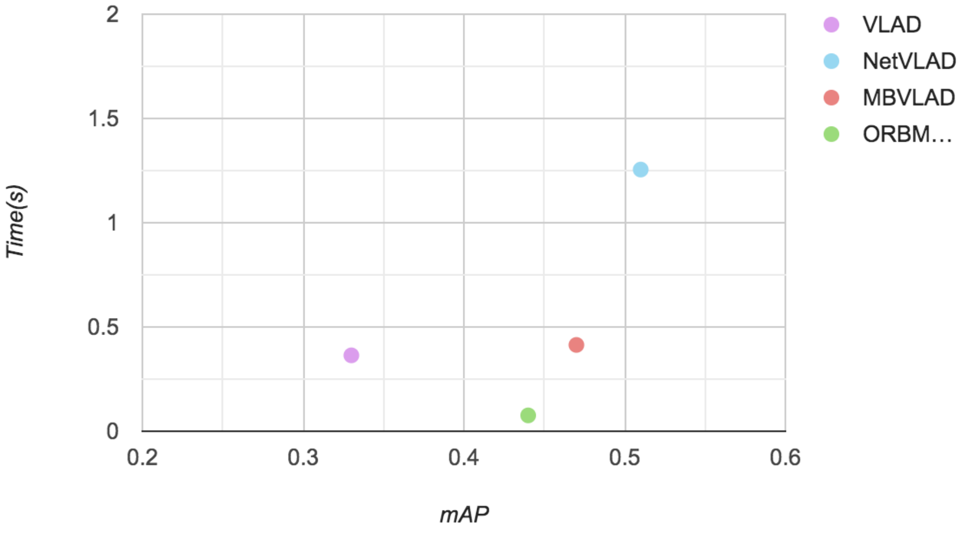


Figure 4.6: Time needed for computation. For Oxford dataset [102], the mAP is shown against the time it takes to analyze a single query image.

be able to replace the VLAD method in many computer vision tasks where fast image representation is a requirement.

Chapter 5

Predicting Relevant Features Using A Hybrid Feature For Visual Place Recognition

In the previous chapters, we made improvements in the retrieval techniques by using MBVLAD and ORB-MBVLAD [5, 2]. However, when the dataset gets larger, those techniques suffer from having to discriminate between features from relevant and non-relevant images. Also, the processing time is very expensive because of the huge number of features. In this chapter, we present our proposed method using a new feature for the extraction and representation phases called a hybrid feature. In the first section, we discuss the motivation and contributions. Then, the design of the proposed method is described. The implementation assessment section shows that we achieve competitive results compared with other baseline methods. Also, our results show a significant improvement when using hybrid features rather than handcrafted models or deep learning methods individually. Finally, we conclude this chapter. Most of this chapter is presented at [4].

5.1 Introduction

5.1.1 Motivation

Visual geolocalization is an important topic in computer vision with numerous applications, including navigation [147] and 3D reconstruction [74]. This study investigates the issue of visual geolocalization at the urban scale. Indirect visual geolocalization approaches take visual geolocalization as an image retrieval issue and are one possible solution. A common goal of indirect visual geolocalization techniques is to choose an image in a reference dataset that most closely matches an input image. For many years, visual geolocalization tasks have relied heavily on Content Based Image Retrieval (CBIR) models that use manually-crafted features [146, 147, 63]. Using a neural network, modern visual geolocalization techniques display images in an embedding space that accurately depicts the proximity of their locations and may be utilized for retrieval. However, the vast majority of studies on visual geolocalization tasks have only covered a localized, neighborhood-sized region. To this end, we have focused our studies on large-scale (e.g., cities). The high quantity of geo-tagged images needed for retrieval and recognition, however, poses a challenge when it comes to figuring out how to put them to good use in the context of training. In the state-of-the-art, most works use the technique of triplet loss learning for training [8, 50]. Those works use negative examples across the training database [8]. This way is expensive in terms of computing. Some works train only small samples of images. This can reduce the training timebut they still result in less effective use of the data. Other works [63, 55, 8] use handcrafted methods and cast the problem as a classification task. They use both positive and negative samples of training. However, in this way, the training can work effectively, depending on the quality of the image representation. The representation should not only be robust against photometric and geometric changes but also have a high discriminative level. We approach the work of solving the image geolocalization problem as a classification task [63, 55]. In this task, image representation is an important phase. Kim et al. [63] proposed using handcrafted representations to extract features. Their work is good in scale and rotation for recognizing a place, but they achieve minimally accurate results when the dataset is large due to appearance changes between the input image and the reference dataset. In the last decade, deep learning representations have been successfully used in many computer vision applications, especially image classification [67]. Because of the multi-layer networks on CNNs that help to extract high-level features from the input image, deep-learning representation methods can extract more comprehensive features. In this way, extracted features using deep learning methods make representation more robust, especially for appearance changes. However, most of the deep learning representations are suitable only for image classification tasks and not for retrieval and recognition tasks. In some cases, fully connected representation is used in visual place recognition tasks. To make robust methods in the presence of occlusions that lack invariance in translation and scale, this representation is computationally expensive due to the large number of parameters required. We offer a novel image representation approach that can extract several patches from an image as a solution to the issue of high computational costs in deep learning representation. This increases the discriminative power of image representation without requiring a huge set of parameters. We do this by combining both machine learning and hand-crafted features. A combination of handcrafted and deep learning features gives a new feature. This hybrid feature outperformed an individual feature in some computer vision applications [91, 137]. In this work, we apply hybrid features to the task of image geolocation and compare the results with individual features. To our knowledge, this is the first time a hybrid feature set has been used for a visual geolocalization task.

5.1.2 Contributions

Our contributions are as follows: (1) We propose an accurate visual geolocalization method that is also effective on a large city scale. Our method can predict good features during the training phase. Not all features can go through the geolocation process. Only features with a high confidence score can be used for the geolocation process. In this way, we achieved good accuracy results, reduced the computational cost, and reduced the number of features compared with baseline methods. (2) We propose a hybrid image representation method for image geolocation systems. Our results show that using the hybrid feature is better by reducing the computational cost during the representation phase with no need to add more parameters.

5.2 Design

5.2.1 Predicting good features from a hybrid method

We aimed to implement an accurate image geolocation system. To get an accurate result, we should have a robust system against geometric and photometric changes that is highly discriminative in large datasets. Our proposed method is a data-driven method that pre-

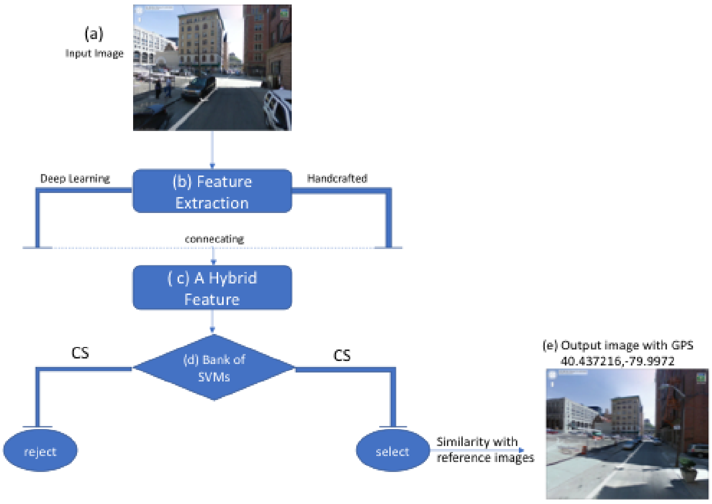


Figure 5.1: (a) input image with no GPS tag. (b) extract feature from input image by using hand crafted and deep learning methods. Then, concatenate features of both methods to get hybrid features set (c). (d) hybrid features go through a bank of SVMs to calculate confidence score CS. Selected features only use for geolocation process. (e) Output is most similar image with GPS.

.

pares the data (features) offline. Figure 5.1 shows our proposed method in the query phase. The input (query) is an image with an unknown GPS. The second step is feature extraction, which is applied by a combination of handcrafted and deep learning models. These two models are combined by concatenating the corresponding features, the result is a hybrid feature. The hybrid feature then goes through a bank of SVM classifiers that already created offline. We then compute the confidence scores of each feature. If a feature has a high score, it is selected as positive (relevant); if not, it is selected as negative. Finally, only the selected features used for computing similarity between the query image and the reference image in the geolocalization process.

5.2.2 Extraction features using the hybrid method

Features play an important role in the image geolocation system; they represent the interesting objects in the street sides of an image. In our pipeline, two types of features are used for extraction: handcrafted features and deep features from CNNs We apply SIFT, VLAD, PB-VLAD, MBVLAD for handcrafted, and AlexNet and SqueezNet for deep local features.

Handcrafted model

We use a variety of handcrafted methods to extract local features from handcrafted models. To create a hybrid feature, we applied each approach once with AlexNet [67] and once with SequeezNet [54].

Scale-invariant feature transforms For image feature extraction, SIFT [77] has been proposed. It converts a single image into a large number of feature vectors. The outcome of the Gaussian function difference is used in scale space. Candidate points with low contrast and edge response points along an edge are removed. Localized critical points are allocated dominant orientations. The important spots will be more stable for recognition and matching as a result of these actions. By evaluating pixels surrounding a radius of the key position and resampling local picture orientation planes, SIFT descriptors resilient to local affine distortion are generated.

Vector of Locally Aggregated Descriptor (VLAD) is a well-known technique for recognizing locations and retrieving images [57]. The descriptor is a low-dimensional vector whose purpose is to serve as a feature for the entire image. D-dimensional descriptors are extracted from affine-invariant detections. Then they were clustered into k centers to create a visual vocabulary. From each descriptor, the residual for each cluster center is calculated. After that, the residual is added together for each cluster, resulting in k D-dimensional aggregate vectors. The Euclidean distance metric is used to compare all vectors in the images.

Per-Bundle VLAD (PBVLAD) [63] combines a packaged VLAD with maximally stable external regions (MSER). In other words, MSER was used to identify areas, followed by SIFT feature descriptions within the detected regions. Then, each region/bundle was described as a fixed-sized VLAD.

Mini Batch VLAD. (MBVLAD) [5]is an extension of VLAD. The local feature descriptors are extracted from an image using a dictionary in the original VLAD. The dictionary is built using a k-means clustering method; in MBVLAD [5], the clustering method is used to generate a dictionary using a mini-batch k-means. The mini-batch k-means approach [112] distributes small batches of a fixed size in a random order. The clusters are updated with each new sample until they reach convergence.

Deep Learning Methods

This framework can be supported by any type of CNN, such as the well-known AlexNet [67]. The task describes feature extraction using AlexNet [67] and SqueezeNet [54] as two instances. Deep learning model.

AlexNet. AlexNet [67]] is an 8-layer deep convolutional neural network. The network was trained with over 1 million images from the ImageNet database. It learned abundant feature expressions from a wide range of images on the network. The image input size for the network is 227x227.

SqueezeNet. SqueezeNet architecture [54], is a small architecture of CNN. It has few parameters and achieves the same accuracy as

AlexNet [67]. This architecture is trained on ImageNet. SqueezeNet architecture follows a strategy of replacing 3x3 filters with 1x1 filters. Moreover, it preserves large activation maps by downsampling late in the network [54]. Then these strategies are packed into a fire module that has a set of 1x1 filters in the squeeze convolution layer and a mix of 1x1 and 3x3 filters in the expand layer. The parameters are about 1,248,424, with a model size of 4.78 MB. In our method, we remove the final softmax layer for both AlexNet and SqueezeNet, as we do not want to classify features by them.

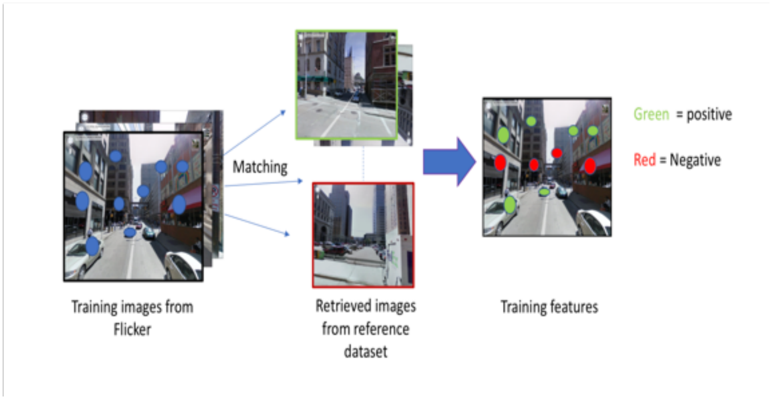

Matching
Training images from Flicker
Retrieved images from reference dataset
Training features
Green = positive
Red = Negative

Figure 5.2: Our offline phase. We extract features from training images that collected from Flicker API. Each query image from the training images, has 100 retrieved images from reference dataset. Set ground truth and false positive images by taking the advantage of geotagged labels.Positive and negative features set depending on the difference between feature in ground truth and feature in false positive. If the difference is greater than 0.8, the feature is assigned as positive; otherwise, it is negative.

.

5.2.3 Feature Selection Technique

After extracting features and feature concatenation, we got a new feature called a hybrid feature. A hybrid feature is a combination of both handcrafted and deep features. In our offline phase, a bank of SVMs

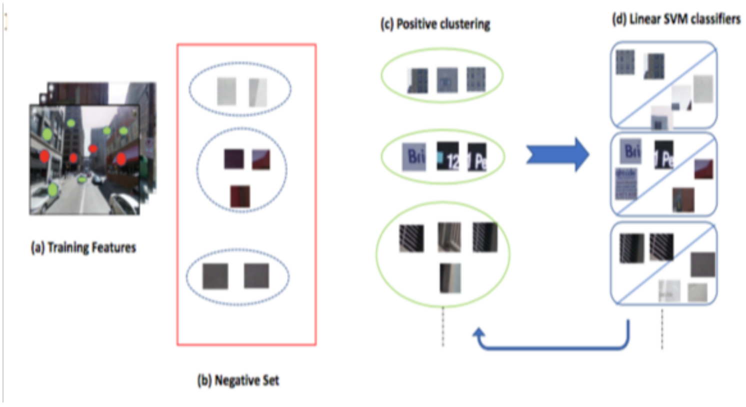


Figure 5.3: Given a set of clusters K, we train a linear SVM classifier as Kim et al. [63]. (a) all features from each training image. A linear model is trained on each cluster in a closed loop. Each classifier uses the firings in its cluster as new positive data for training. We applied iterative hard mining to treat the complexity.

.

was created, which includes positive and negative features. We use SVM with a hard margin to make the relevant and non-relevant features linearly separable. In the feature selection step, hybrid features is classified as positive or negative by computing the confidence score CS. Features with high weight will only be used for the visual geolocation process. In this way, we got a good accuracy while using a few numbers of features. This helps with computational costs and search time as well.

5.2.4 Learned Classifier in Offline Phase

Our goal is to predict positive and negative features by training a linear support vector machine (SVM) classifier offline. In the previous works, this step was used on input images in query time, and the weight vector is applied as a new query image representation. However, the computational cost will be very expensive because prediction features require training a new classifier for each query. Hence, we follow other works that learn classifiers for each place in the database offline [63, 55, 8].

In this section, we first introduced how we automatically generate training data (features) for a learning classifier. Then, a bank of SVM classifiers is introduced.

Generate features We generate features automatically for each image in Reference dataset R for learning a classifier task. We collected new geotagged images S from social media platform which has the same GPS locations in each image in reference dataset R. Given x is an image descriptor for each image x in the reference dataset R. The representation is a hybrid feature h. Our goal is to learn F_i which i^* = $arg_i \max F_i(g)$ To generate training features for learning classifiers, we first rank the top 100 images for each image in S by computing the similarity with images in R. Also, we use geometric verification [36] to identify the correct location between R images and images from S. We employ the geotags to build the False positive R_n images and ground truth images R_p for each image i. The False-positive set R_n is images that are in the ranking images and far away from the location of image i. The ground truth set R_p is images near given GPS location. Figure 6.2 shows the offline phase. For this purpose, we collected 720 Flickr geotagged images as our training image set S. The training images S cover the same region as the reference images R [147]. Next, we follow the same process as Kim et al. [63, 55]; for each image in the training dataset S, we define positive images from the reference dataset R_p if the image is within 50m of the given GPS location and passes geometric verification by using RANSAC with respect to S. To verify negative images R_n, we take the reference image with the smallest distance to query image in the training dataset S, which is about 225m from the given GPS location which is about 225m from the given GPS location. After feature training, we realized that some features in different objects appear in the same class. That is because a single classifier applied to a large dataset affects the appearance variation. To To solve this problem, we create a bank of SVM classifiers [63] and apply the bottom-up clustering technique. The idea is to get clusters of positive and negative examples that are most consistent between labels and appearances. Also, in each cluster, we obtain a trained bank of linear SVM classifiers.

We feed features obtained from input (query) images into the bank of linear SVM classifiers in the query pipeline. We figure the confidence score CS to predict the useful features prior to the geolocalization process. In simple terms, we follow some works that used the discriminativeness technique of the classifier [63]. We define the discriminativeness of a feature as the ratio of the number of firings on its cluster to the number of firings on the entire training set. We consider all SVM scores above 1 to be firings. As a result, we keep only the features with high confidence scores for the visual geolocalization task [63]. Figure 5.3 shows the training of SVM classifiers in the offline phase.

5.3 Implementation Assessment

5.3.1 Implementation details

The work was implemented on a workstation outfitted with a 32GB RAM, i7 CPU running on twelve threads, 144GB of swap memory, and a Titan Xp GPU. Python 3.5 is used as the development environment. We run a rough vocabulary of 128 visual words and 16,384 dimensions through our constructed models for VLAD, PBVLAD, and MB-VLAD. We use a visual vocabulary of 16 terms as well. Pre-trained

Table 5.1: Quantitative evaluations using evaluation metrics: recall @1 and @5. All results show the performance of the evaluation dataset from Google Street View that was provided by [147]. The results applied to the reference dataset from [147] that includes street view images from Pittsburgh City. Higher values are better with Recall @1 and @5 metrics. Note that the results were within 25 and 15 meters.

Methods	25m		15m	
	R@1	R@5	R@1	R@5
SIFT [146]	43.26	48.46	40.11	44.81
PBVLAD [63]	65.17	68.39	62.71	65.33
NetVLAD [8]	78.58	75.63	72.25	72.08
PatchNetVLAD [50]	67.54	70.28	65.25	67.42
Our SIFT	53.60	50.33	49.85	49.21
Our VLAD	56.76	57.34	54.77	51.89
Our PBVLAD	59.43	56.12	56.52	54.10
Our MBVLAD	62.74	62.15	60.81	59.04
Our AlexNet	64.53	63.10	61.38	60.77
Our SequeezNet	65.42	64.95	62.49	61.25
Our Hybrid SIFT+AlexNet	66.20	65.41	63.40	63.11
Our Hybrid SIFT+SqueezNet	66.47	66.13	65.51	64.06
Our Hybrid VLAD+AlexNet	67.02	66.58	65.80	65.21
Our Hybrid VLAD+SqueezNet	68.48	67.71	65.86	65.15
Our Hybrid MBVLAD+AlexNet	70.42	69.43	67.17	66.22
Our Hybrid MBVLAD+SqueezNet	78.40	74.32	72.51	71.13

Table 5.2: Quantitative evaluations using evaluation metrics Recall @1, Recall @5, and Recall @10. All results show the performance of the evaluation dataset from the Pittsburgh 30k dataset [8], and the results for all methods within 25 meters. The performance applied to the reference dataset is from [147] that covered Pittsburgh city. Note that NetVLAD and PatchNetVLAD trained using different datasets [50], and the reference dataset is not the same as our methods.

Method	Recall @1	Recall @5	Recall @10
SIFT (2010) [146]	32.02	34.33	37.62
PBVLAD (2015) [63]	40.26	44.72	46.81
NetVLAD (2016)[8]	83.5	91.3	94.0
PatchNetVLAD (2021) [50]	88.6	94.5	95.9
Our MBVLAD+AlexNet	65.54	69.38	70.21
Our MBVLAD+SqueezNet	78.16	79.63	80.94

models in deep learning use Keras and TensorFlow to create AlexNet and SqueezeNet architectures. Here, we get rid of the very last softmax layer that was ever employed for classification.

5.3.2 Evaluation dataset

The reference image set R is a Google Street View dataset obtained by Zamir and Shah [147]. Each image in this dataset is labeled with GPS and compass coordinates, and all images were taken in outdoor environments under similar physical conditions. In our experiment, the reference dataset size comprises approximately 27,500 images covering only Pittsburgh, PA. We partition the dataset into 23,500, 2,000, and 2,000 images for training, validation, and testing, respectively.

We compare our methods with different state-of-the-art visual geolocation methods, including: SIFT [146], VLAD [57], PBVLAD [63], NetVLAD [8], and PatchNetVLAD [50]. Moreover, we test our proposed method by applying different feature extraction methods: SIFT [77], VLAD [57], PBVLAD [63], MBVLAD [5], SeqeezeNet [54], and AlexNet [67]. Furthermore, we test our method using our hybrid features: SIFT+AlexNet, SIFT+SeqeezeNet, VLAD+AlexNet ,VLADSeqeezeNet, MBVLAD+AlexNet, and MBVLAD+SeqeezeNet.

5.3.3 Experimental results

We use the standard visual geolocation evaluation technique [63, 8, 50] to provide a numeric assessment of the obtained data. It is often accepted that recall is the most discriminatory metric. Typically, an image retrieval system will choose the top k (where k is an integer between 1 and 10) choices and then check to see whether any of them are within a localization tolerance. Therefore, we test our approaches using a variety of top recall values and within a couple of error thresholds. The mean estimate error is used in conjunction with the Recall@N metric to determine the accuracy. If one assessment is within 25 meters and the other is within 15 meters of the query location, we consider the geolocalization to be successful. The recall method is used for the top five and the top one. To determine whether an image is in the top-1 score, we look to see if it is the first prediction. For the top-5 score, we look to see whether the target image is among the five most likely predictions. To test the efficacy of our suggested approach, we use two data sets.

The initial collection of data consists of two thousand images captured from Google's Street View service [147]. Table 2 displays the numerical outcomes as determined by the assessment criteria Recall @1, Recall @5, and Recall @10. The Pittsburgh 30k dataset cited by [8] was used for all analyses.
The numerical results of our methods against several benchmark localisation solutions – SIFT [146], VLAD [57], PBVLAD [63], NetVLAD [8] and PatchNetVLAD [50] – are shown in Table 1. Our proposed method Hybrid MBVLAD+SequeezNet shows competitive results compared with the rest of the methods. On the other hand, as shown in table 5.1, our solution incorporating Hybrid methods outperforms the solution using individual methods: handcrafted models or deep learning methods. Moreover, the quantitative metrics show that our methods and all the state-of-the-art methods improve when the error threshold is greater. This indicates that all methods still have difficulty retrieving an accurate target image in terms of its very close

geolocation to the query image. Table 5.2 shows the numerical results measured by using evaluation metrics Recall @1, Recall @5, and Recall @10 of the state of the art compared with our best hybrid methods, MBVLAD+AlexNet and MBVLAD+SeqeezNet. All results were evaluated on the dataset from the Pittsburgh 30k dataset [8]. Moreover, the results for all methods were within 25 meters. In this table 5.2, NetVLAD and PatchNetVLAD are trained using different datasets, and the reference dataset also differs from our method and the rest of the state of the art. In conclusion, NetVLAD is outperformed by the other methods in 5.1 and 5.2, due to the strong dataset that the network trained on.

We also use a line chart to graphically represent numerical data. The goal of the line charts is to graphically represent the distribution of numerical data, to highlight differences across approaches, and to demonstrate how close our methods are to state-of-the-art algorithms. In 5.4, we show the performance of our prediction method when using traditional representation methods. We compare our results with other VPR methods that used traditional methods such as SIFT [77] and PBVLAD [63]. Figure 5.5 shows the Recall@N performance of our prediction feature method when deep learning representation methods are applied. The figure also shows the comparison of our performance with other VPR methods, those using deep learning NetVLAD [8] and PatchNetVLAD [50]. In figure 5.6, the chart indicates that our predicting method uses traditional, deep learning, and hybrid representation methods. Finally, figure 5.7 shows the effectiveness of our proposed method by comparing different hybrid features.

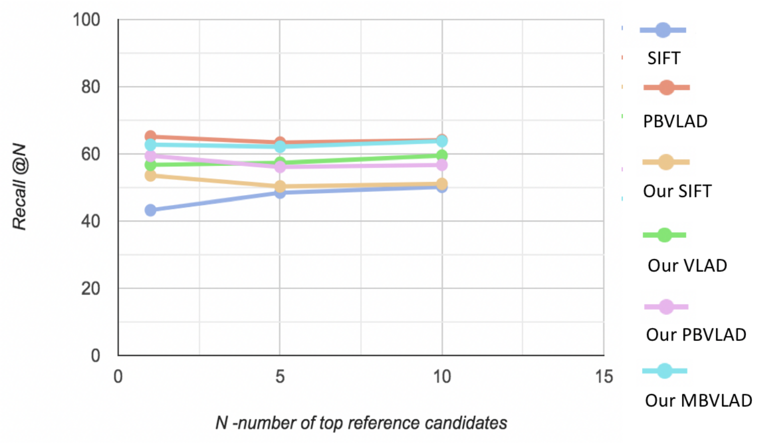


Figure 5.4: Comparison with state-of-the-art. The Recall@N performance of traditional representation methods (SIFT [146] and PBVLAD [63] compared to our representation methods (traditional methods with good feature predictions). All results show the performance of the evaluation dataset from pittsburgh 30k dataset [8], and the results for all methods within a 25 meter.

.

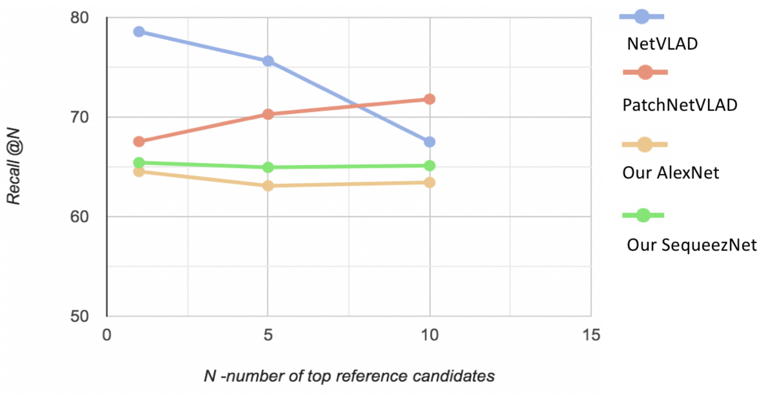


Figure 5.5: Comparison with state-of-the-art. The Recall@N performance of deep learning representation methods (NetVLAD [8] and PatchNetVLAD [50] compared to our representation methods (AlexNet [67] and SequeezNet [54] methods with good feature prediction). All results show the performance of the evaluation dataset from pittsburgh 30k dataset [8], and the results for all methods within a 25 meter.

.

5.4 Conclusion

We introduce a new visual geolocalization method at a large city scale.

Our method is a data driven method that can reduce the number of

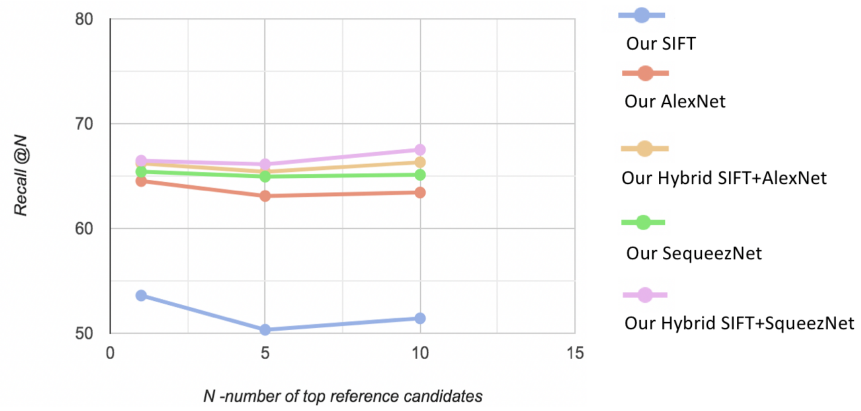


Figure 5.6: Comparison of the Recall@N performance of our predicting method when using the traditional method SIFT [77], deep learning representation methods (our AlexNet and our SequeezNet), and our hybrid models. All results show the performance of the evaluation dataset from pittsburgh 30k dataset [8], and the results for all methods within a 25 meter.

.

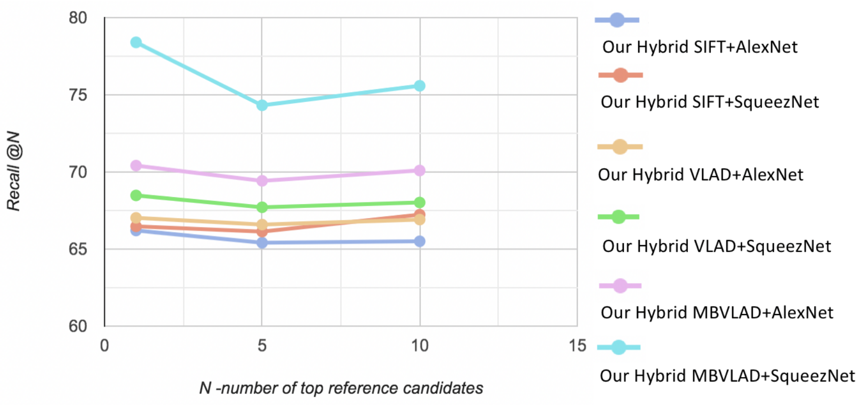


Figure 5.7: Comparison of the Recall@N performance of our predicting method when using several hybrid methods. All results show the performance of the evaluation dataset from pittsburgh 30k dataset [8], and the results for all methods within a 25 meter.

.

features prior to the geolocation process by predicting relevant features during training time. In the image representation phase, we represent images with a hybrid feature set that incorporates handcrafted and deep-learning models. We show that our method improves over the state-of-the-art results for visual geolocalization and reduces the computational cost. We evaluate the effectiveness of our approach as it relates to accuracy. We achieved good accuracy while reducing the number of features. Also, our results show an improvement compared with the use of only handcrafted models or deep learning methods.

Chapter 6

Efficient Visual Place Recognition System by Predicting Unique Features

This chapter will present the last contribution to this thesis. This chapter is structured as follows: First, an introduction is provided, describing concerns and shortcomings with the then-current state-of-the-art. Following that, we describe our method, including the dataset, our method structure and training pipeline, and all training specifics. Following that, our approach is evaluated in comparison to the state-of-the-art. Finally, we provide a conclusion that discusses the method's weaknesses and limits, as well as prospective enhancements. Note that most of this chapter was published in [3].

6.1 Introduction

6.1.1 Motivation

To perform higher-level tasks like planning and navigation, a robot must always have an accurate assessment of its location and orientation in relation to the environment. VPR approaches typically presume that the appearance remains constant from the moment the map (reference) is produced until the time the robot needs to locate itself. However, as the robot's operating life span expands, the look of the environment changes. This presents a significant difficulty for VPR approaches since the basic premise of static appearance is broken owing to constant changes in the environment, such as weather, time of day, building sites, upgrading of facades and billboards, and so on.

A prominent way to deal with environmental change resilience is to demand the selection of positive and negative examples for each training image [8, 55, 43]. Positive and/or negative examples are presented to the model for each training sample. The two target points mandate that the learned representation for the training sample is near that of positive instances and distant from that of negative examples, according to a metric. A positive example in the context of VPR is an image of the same location as the training sample, whereas a negative example is an image of a different location. The cutting-edge research forces the system to learn examples of both positive and negative. However, only a few studies propose avoiding learning from negative examples by removing them from the reference dataset.

By doing selection features by taking advantage of GPS data, CRN [55] and PBVLAD [63] proposed that the collection of positive examples be defined as images that are within 50 meters of the training query's GPS tag. Given the possibility of taking photos from the same GPS coordinates while pointing the camera in different directions, the positive candidates are narrowed using geometric verification. For the negative examples, the authors replicate the picture geo-localization procedure inside the training batch, and for each iteration, they select the top retrieved image that is at least 225 meters distant from the GPS position of the training query as a negative candidate. The negatives are likewise selected at random from the batch. Unfortunately, the substantial amount of time required to compute the similarity between each feature in the training image and all reference datasets is its bottleneck. In NetVLAD, for each training sample, the selection takes into account all of the negative cases. A simple calculation of all negatives is impractical because it would have each query doing a forward pass on all database images. Furthermore, many negative instances would have a small impact on the decision, so analyzing them would be a waste of time.

Even though recent algorithms [63, 8, 55] demonstrate that forcing the methods to train negative examples leads to correct localization, they are not supported by any fundamental theory in which removing negative features can lead to bad localization. In this chapter, our goal is to propose a VPR method by avoiding confusing features and evaluating its performance in terms of robustness and efficiency.

6.1.2 Contributions

We improve on the method of the state of the art to better deal with the problem by proposing a method for automatically selecting such "mismatching features" and demonstrate that removing them from the query image can significantly improve the place recognition performance. To this end, we predict good features by taking advantage of GPS tags prior to the geolocalization process. In this way, we discriminate against images to find the place correctly while reducing the number of features. In addition, we propose a new dataset based on images collected from the social media platform. We split the dataset for training and testing. To generate features automatically, we use training set and test set images to evaluate the robustness of our method with hard images.

6.2 Design

The goal of Visual Place Recognition is to find the same place in a large geotagged dataset of outdoor images where a given query image has an unknown GPS. To that end, each feature in the query image should be compared to each feature in the reference dataset's images. We propose modifying this task so that the number of features in the query image is reduced before computing similarity. In particular, we use GPS data to predict good features and learn from positive and negative examples. The problem then becomes identifying the confusing features and avoiding their use without affecting learning image discrimination. Another challenge is that, despite minimizing the number of features, the proposed method should be efficient. In other words, performance should be improved while reducing the cost of computational and time-consuming tasks. In the following sections, we go over our method in significant detail.

6.2.1 overview method

Generally, following the success of current place recognition systems [8, 50], we recast location recognition as image retrieval. The query image with an unknown location is used to visually search a large geotagged image database, with the locations of top-ranked images used as possibilities for the query's location. This is often accomplished by creating a function f that serves as the "image representation extractor," producing a fixed-size vector f given an image $I_i(Ii)$. The function is used to extract the representations for the complete database I_i, which can be done offline, and the query image representation $f(q)$, which may be done online. The visual search is conducted by locating the database picture that is the closest to the query, either exactly or by rapid approximation nearest neighbor search, by ordering photos based on the Euclidean distance d(q,Ii) between f(q) and f(Ii) (Ii). In our method, we input an image, and after extracting features from the image, we compare the similarity score between positive and negative features in the bank of SVMs that have already been built offline. If a feature has a high score, it will only be used for the final step, which is computing the similarity between the positive feature on the query image and all reference dataset images. By doing this, we use only a small number of features, which reduces the computational cost in the online phase.

6.2.2 Problem Formulation

Visual Place Recognition methods use image retrieval techniques. Indeed, image retrieval is a "learning-to-rank" challenge, which is a type of learning image descriptor that properly depicts similarity from the perspective of a distance function. For representation learning via ranking, selecting positive p^+ and negative examples p^- for each training image q is required. The method requires that the learned representation for the training sample be close to that of positive examples p^+ and far from that of negative ones p^+. In Kim et al. and

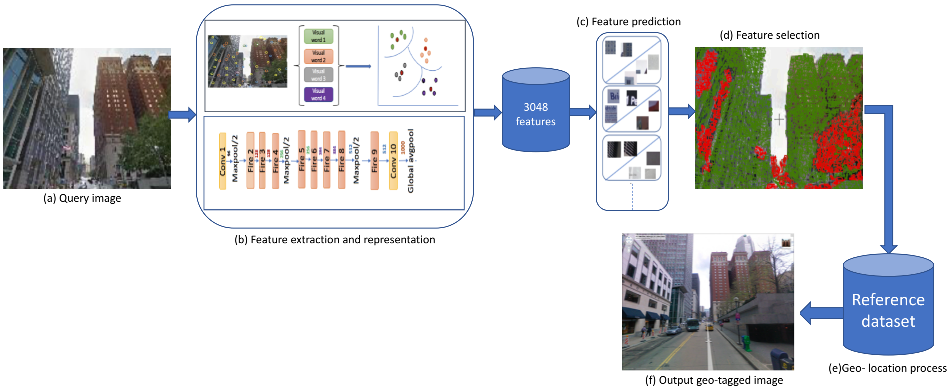


Figure 6.1: our proposed method. (a) An input image with unknowing GPS tag. (b) features extracted from the input image. (c) A bank of SVM classifier with positive and negative features. (d) Computing confidence score if high the feature labelled as positive, else negative. (e) Positive features only used for geo-location process. (f) The output image with geo-tag.

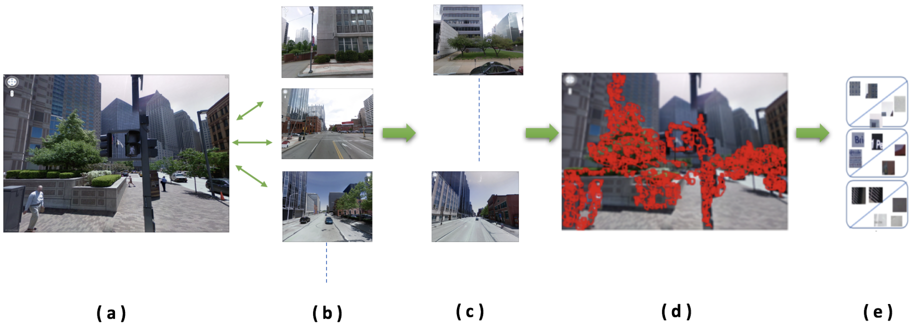


Figure 6.2: Examples of non-unique objects found by finding local features in the original image.

NetVlad, the VPR models are trained by feeding q, p^+ and p^-. p^+. The positive examples are obtained by ranking all images from the reference dataset I within a small geographical distance from query q and taking images with the highest visual similarity. All negative images p^- are far away from the query q geolocation. In selecting the positive and negative examples, the goal is to learn a distance function d. The distance $d_{(q,p^{+})}$ between the training query q and the positive will be smaller than the distance d between the query q and all negative images. To minimize the distance, In our proposed method, we modify this by ranking the list of mismatching images M with the query image. The negative examples are 50 meters away from the query image. If the feature from q has a high similarity score with the feature from M, the feature is labeled as negative. otherwise, positive.

6.2.3 Automatic generating features

Locations in city-street image collections contain a substantial quantity of features on objects such as trees or road markers, which are not useful for identifying a specific location because they occur often around the city. This is a big issue since such features clog the visual word vectors and can generate substantial ambiguity between various locations. This section focuses on automatically finding such locations in order to remedy this issue. To do this, we exploit the fact that an image of a certain site should not match well with other images from distant locations.

For each image i in the training dataset I_q that is already collected from social media platforms, the top n "confusing" images were retrieved from the geotagged database I_t. This is accomplished by retrieving top-matched images using the feature matching similarity equation. We exclude images at places closer than (parameter) meters from I to guarantee that the retrieved images do not include the same visual scene. Feature matching equation $f(p_q,I_r) = maxM(p_q,p_r)$, similarty equation $sim(I_q,I_r) = sum_{p_q} f(p_q,I_r)$ After we get the top n mis-matching images, we want to automatically generate positive/negative examples of features using just their related

GPS tags. Rather than making assumptions about what features are good and bad for geo-localization, we want to automatically identify them using available data. This allows us to adjust our strategy to different geographical locations. Features with a high matching score will be labeled as negative features; others will be positive. If similarity between features in a training image has a high matching score, the feature will be labeled as a negative feature. Otherwise, it will be positive. This accounts for both user-supplied geo-tag inaccuracies and the fact that huge, symmetric buildings are frequently visible from long distances. If similarity between features in a training image has a high matching score, the feature will be labeled as a negative feature. Otherwise, it will be positive.

After feature training, we realized that some features in different objects appear in the same class. That is because a single classifier applied to a large dataset affects the appearance variation. To solve this problem, we create a bank of SVM classifiers [63] and apply the bottom-up clustering technique. The idea is to get clusters of positive and negative examples that are most consistent between labels and appearances. In each cluster, we obtain a trained bank of linear SVM classifiers ready to be used for prediction in the online phase. Refer to Chapter 3 for detailed details on the training dataset collection.

6.2.4 Online phase

Our proposed method is shown in Figure 6.1. First, query image q extracted features by using a hybrid feature. A hybrid feature is a combination of two image representation methods used for visual place recognition tasks. We applied using SqueezeNet [54] and VLAD [57]. In the offline phase, we first rank the mis-matching images with each query image based on their similarity. We make sure no image is

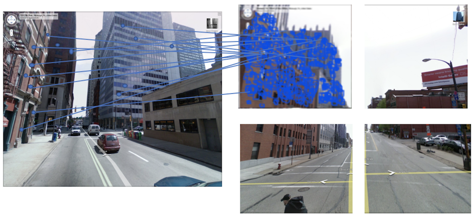


Figure 6.3: Detection of place-specific confusing features. Left is the query image, Features in each database image are matched with features of similar images at geospatially far away locations (illustration of matches to only one image is shown). Right image from reference dataset and detecting points. Note that we spilt the image to four parts to illustrate how detect confusing features.

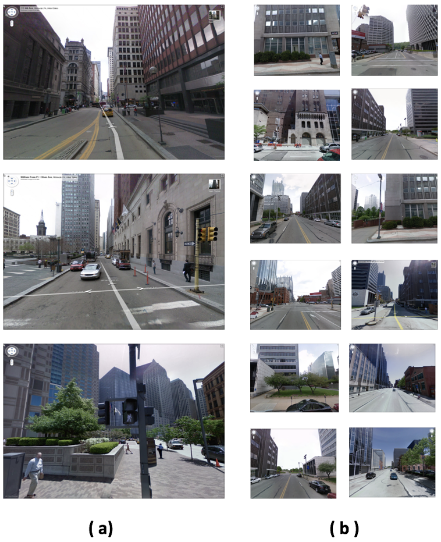


Figure 6.4: In original image (a) frequently mismatched to similar images of different places shown in (b).

similar to the query or at the same GPS distance. We then compute features to define positive and negative examples by computing feature scores. If the score is low with images in the ranking, then it is labeled as negative; otherwise, it is positive. In this way, we reduce the computational process by computing only the score in negative examples. After we get positive and negative features, we cluster them as groups. Then learn SVM to prepare features for online use. To create a bank of SVMs for predicting features for query images, training is done offline. Following the work of Kim et al. [63], we solve the problem of including such category-level place analysis in the current framework to further improve the place recognition performance.

6.3 Experimental Assessment

We provide quantitative and qualitative evaluations to validate our method, and our solution predicts good features for the geolocation process by ranking mis-match images and minimizing the similarity distances between q and p^{-}. The Recall@N metric is applied to evaluate all datasets, and a query image is correctly localized if at least one of the top N images corresponds to the ground truth. To visualize the results, we first modify the value of N and then compute the recall as the percentage of successfully localized query images. For all datasets, including Pittsburgh 250k, Google Street View, and Flickr images, we define an accurate localization as one that falls within 25 meters of the usual ground truth.

6.3.1 implementation details

The entire project is carried out on a workstation equipped with an i7 CPU with six cores (12 threads), 32GB of RAM, 144GB of swap memory, and a Titan Xp GPU. Python 3.5 is the development software. The parameters employed on handmade approaches in the extraction phase are: a coarse vocabulary of 128 visual words and 16,384-dimensionality; and a visual vocabulary of 16 words. Other large codebases and libraries used were OpenCV [12] (for image-related operations) and Scikit-learn [100] (for scientific computations). The AlexNet and SqueezeNet architectures are implemented by Keras and TensorFlow. We eliminate the last softmax layer used for classification jobs because we only utilize it for extraction. Each image has a resolution of 640x480 pixels.

Evaluation matrecs. We follow the standard evaluation procedure [8, 63]. The localization is deemed correct if at least one of the top N retrieved images is within d =25 meters from the query. Recall@ N is the percentage of correctly localized queries for different values of N.

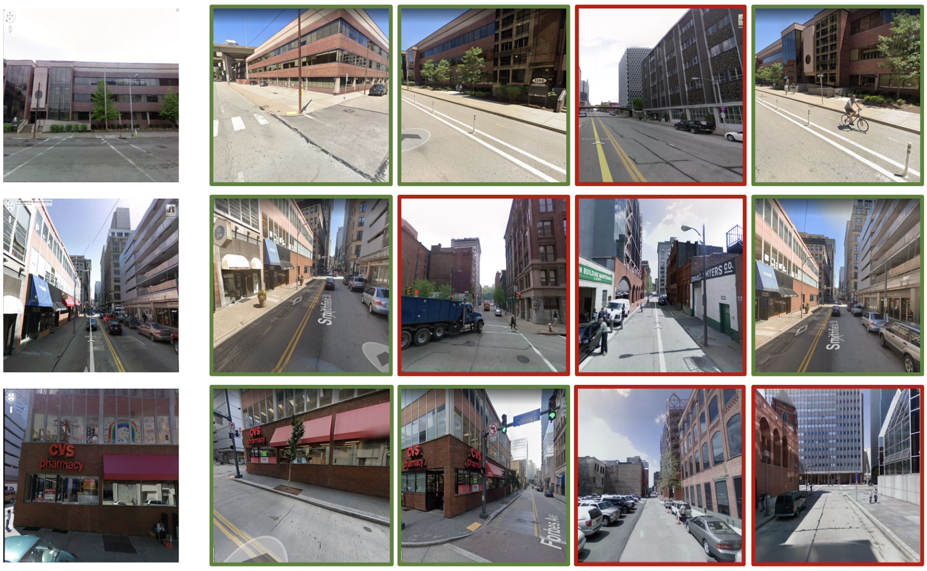


Figure 6.5: Qualitative Outcomes. This is evidence that the suggested approach successfully recovers the corresponding reference image. Example retrieval results on Flicker. The Flicker images match example is particularly challenging, with severe viewpoint shift, occlusions and objects at the street sides such as cars, people. From left to right, query image, and then retrieved results. Green and red borders indicate correct and incorrect retrieved results, respectively.

6.3.2 The state of the art algorithms for comparison

6.3.3 Evaluation dataset

We trained our method using Pittsburgh 250k. We used the corresponding training and validation sets for this dataset, namely, the Pittsburgh 250K training and validation set. The datasets collectively cover a wide range of perspective- and appearance-changing situations, due in part to major differences in the collecting technique, such as cars and general crowdsourcing. Different times of day, varied weather, and seasonal changes produce specific looks. Furthermore, we test our method with three datasets. Pittsburgh 250k test, Google Street View dataset, and images from Flickr. More details about datasets are in Chapter 2.

6.3.4 Evaluation tools

Numerical metrics: To quantitatively assess the results, we follow the same evaluation as in [8, 55, 50]. Three recalls are used. r1, r5, and recall 10. To quantitatively assess the visual geolocation results, we followed the standard visual geolocation procedure [63, 8, 50]. The recall metric is often used as the most discriminative metric. It is typical for an image-retrieval system to choose the top-k $(1 <= k <= 10)$ ranked candidates and evaluate whether any of the candidates lie within a tolerance radius for localisation. Thus, we evaluated our methods with different levels of recall and within the error threshold. In this case, the accuracy was computed by means of the estimation error, the distance between the true geolocation of the query image and the predicted one using the Recall@N metric. We considered the geolocalisation successful if it was within 25 meters, and another evaluation was within 15 meters, in the vicinity of its true location. We applied recall for the top 1, top 5, and top 10. For the top-1 score, we checked if the target image was the first one of the predictions. For the top-5 score, we checked if the target image was one of the top five predictions, meaning the five with the highest probabilities. For the top-10 score, the system checked whether the target image was one of the top ten predictions with the highest probabilities.

Descriptive statistics: We use a line chart to graphically represent numerical data. The goal of the line charts is to graphically represent the distribution of numerical data, to highlight differences across approaches, and to demonstrate how close our methods are to state-of-the-art algorithms.

Significance analysis: To analyze the effectiveness of our method, we compare it with the following variations: 1) Instead of a positive feature selection, the geolocation technique employs a hard negative one. 2) Both negative and positive features are selected in the geolocation process. We further study the robustness of our approach to the choice of image representations. We applied some traditional and deep learning representations for the extraction phase and compared our method's performances each time. Furthermore, we analyze our strategy in terms of computing time requirements to assess its efficiency. We compare our method to other state-of-the-art methods by determining the number of queries that each method can handle per second. In addition, we compare our method with various feature selections and compute the time and performance requirements. We also evaluate our method in terms of an algorithm's time complexity as a function of its input size. In particular, we measure how the temporal complexity of an algorithm increases with increasing input size. To do this, we apply an asymptotic time analysis calculation.

Qualitative analysis: We offer visual evaluations. The results are shown to demonstrate the query images and the accurate retrieval for clarity. If the output image is correct, the outcome should be a green border; otherwise, a red border. This makes the comparison precise and straightforward.

Table 6.1: Quantitative evaluations using evaluation metrics: Recall @1, @5 and @10. All results show the performance of the evaluation dataset from the Pittsburg 250k dataset that was provided from [8]. Higher values are better with each Recall @N metrics. Note that the results are within 25 meters.

Methods	R@1	R@5	R@10
NetVLAD	86.0	93.2	95.1
CRN	85.5	93.5	95.5
SFRS	90.7	96.4	97.6
PatchNetVLAD	88.7	94.5	95.9
Ours	89.1	94.6	96.3

Table 6.2: Quantitative evaluations using evaluation metrics: Recall @1, @5 and @10. All results show the performance of the evaluation dataset that was provided by our collection dataset from social media. Higher values are better with each Recall @N metrics. Note that the results are within 25 meters.

Methods	R@1	R@5	R@10
PBVLAD	68.1	70.3	70.9
NetVLAD	77.5	78.2	80.3
CRN	73.2	73.8	74.2
SFRS	75.1	76.4	78.6
PatchNetVLAD	75.3	77.3	79..1
Ours	77.3	78.6	80.9

Table 6.3: Quantitative evaluations using evaluation metrics: Recall @1, @5 and @10. All results show the performance of evaluation dataset from Google Street View dataset that provided from [146]. Higher values are better with each Recall @N metrics. Note that the results within 25 meters.

Methods	R@1	R@5	R@10
PBVLAD	50.5	58.4	60.1
NetVLAD	70.9	75.6	78.2
CRN	68.2	61.7	65.9
SFRS	70.1	74.5	78.7
PatchNetVLAD	69.4	74.1	76.7
Ours	71.3	76.2	80.1

Table 6.4: Quantitative evaluations using the evaluation metric Recall @1 for our ablation studies. All results show the performance of the evaluation dataset from the Pittsburg 250k dataset that was provided by [8]. Higher values are better with each Recall @1 metric. Note that the results were within 25 meters. Other evaluations for other datasets and Recall @N, see Appendix.

Methods	R@1	R@5	R@10
P^+ selected	89.1	94.6	96.3
N^- selected	58.3	59.1	62.4
Selected all	75.9	77.5	79.1

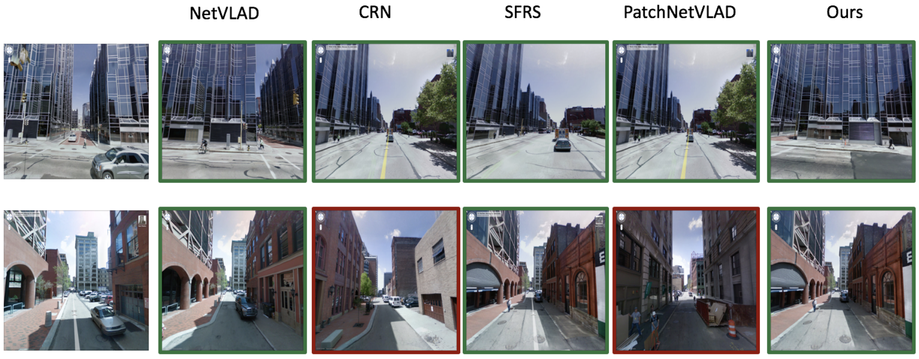


Figure 6.6: Qualitative Results. In these examples, our proposed method successfully retrieves the matching reference image, while CRN and PatchNetVLAD produce incorrect place matches.

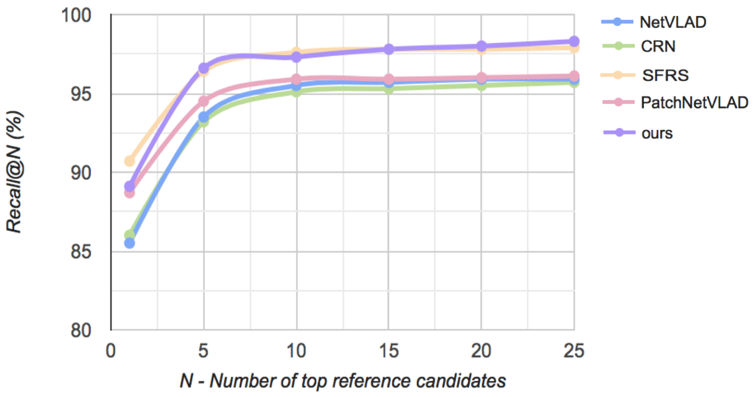


Figure 6.7: Comparison with state-of-the-art. the Recall@N performance of Ours (predicted) compared to NetVLAD, CRN, SFRS, and PatchNetVLAD on the Pittsburgh 30k test dataset.

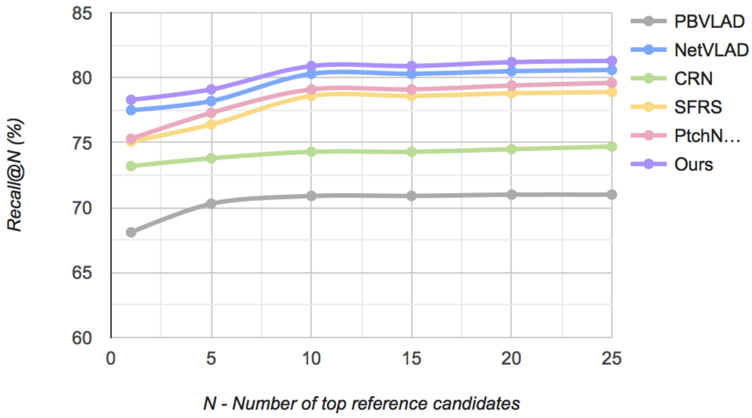


Figure 6.8: Comparison with state-of-the-art. the Recall@N performance of ours (predicted) compared to PBVLAD, NetVLAD, CRN, SFRS, and PatchNetVLAD on our collected Flicker images.

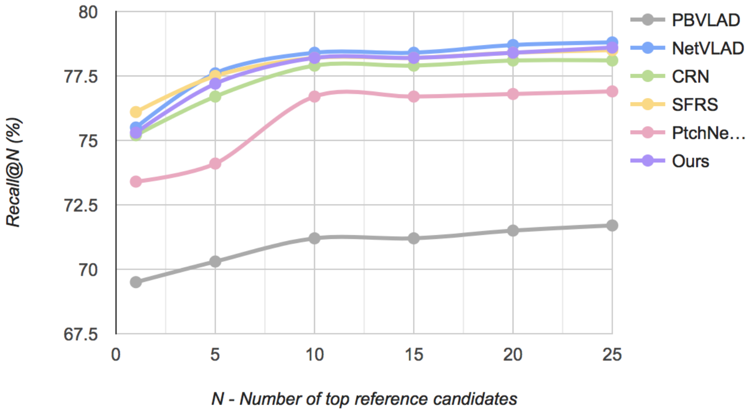


Figure 6.9: Comparison with state-of-the-art. the Recall@N performance of ours (predicted) compared to PBVLAD, NetVLAD, CRN, SFRS, and PatchNetVLAD on our collected Flicker images.

6.3.5 Experimental results

Quantitative Evaluation:

To evaluate the performance of a VPR model, the model must accurately recognize/localize a query image location. Different evaluation approaches might be employed depending on how the place is recognized. The query image is successfully localized for outdoor VPR if it occurs near a defined GPS threshold where the image was captured. The numerical results of the Recall metric with different levels and

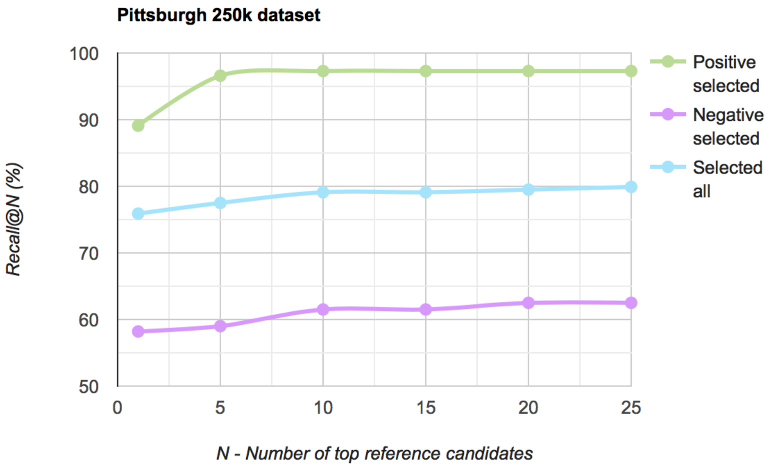


Figure 6.10: Comparison with our different selections. the Recall@N performance of Ours selected compared to negative selected and all selected positive and negative on the Pittsburgh 30k test dataset.

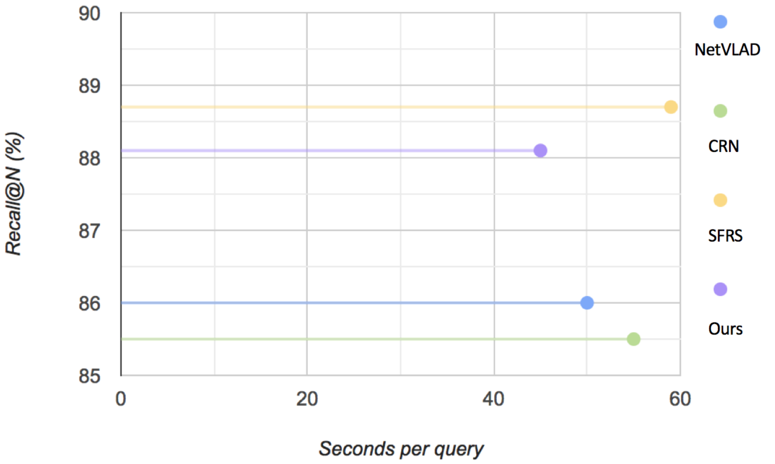


Figure 6.11: Time spent processing a single query image (x-axis) and R@1 (y-axis) is shown below for the Pittsburgh 250k dataset.

within 25 meters are shown in Tables **??** , Table 6.2, and 6.3, along with line charts revealing many statistical details. The purpose of the line charts is to visualize differences among methods and to show how close our methods are to the state-of-the-art algorithms.

The numerical results of our method against several benchmark localisation solutions NetVLAD [8], CRN, SFRS, and PatchNetVLAD [50] – are shown in Table**??**. All the methods were evaluated with the Pittsburgh 250k dataset, and we used R@1, R@5, and R@10 as

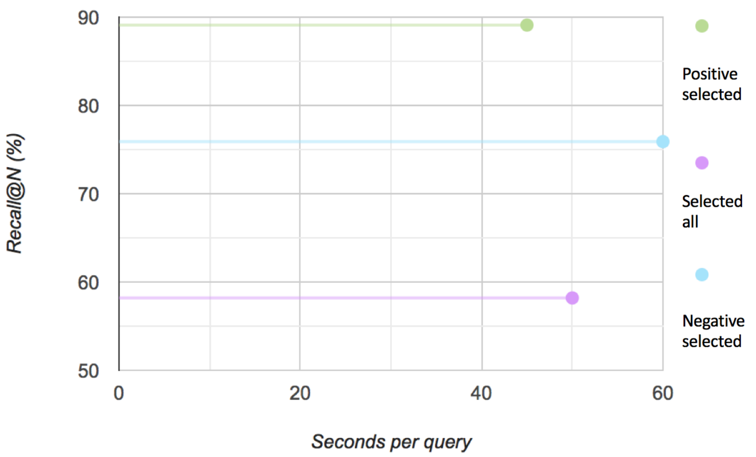


Figure 6.12: Time spent processing a single query image (x-axis) and R@1 (y-axis) is shown below for the Pittsburgh 250k dataset.

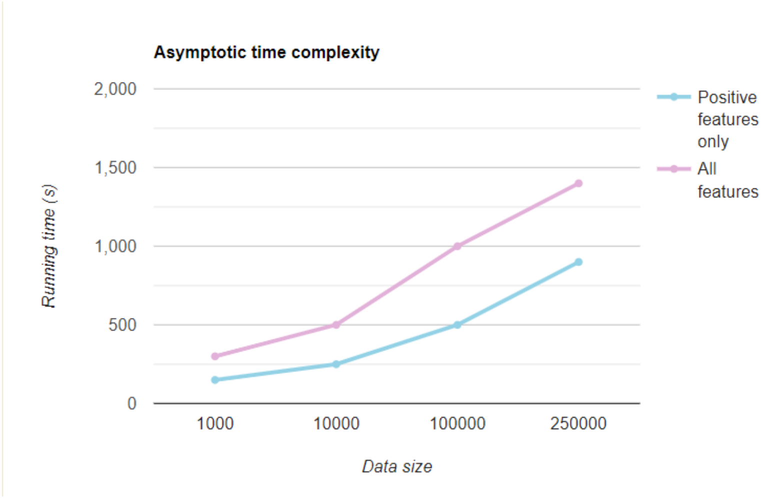


Figure 6.13: An asymptotic analysis for the time complexity of our algorithm with selection and non-selection when the input size increases.

reported by the authors.

Our method outperforms the best performing VPR methods, NetVLAD, CRN, and PatchNetVLAD, on average by 3.1, 3.6, and 0.4, respectively. However, SFRS achieves better performance; this comes at a high computational cost (10 times slower than our best-performing. The line charts in the figure illustrate the Recall calculated by varying N in our approaches and other baseline methods.

Moreover, Table 6.2 contains quantitative comparisons of our proposed method and the baseline methods on Google Street View test datasets. Note that all results shown in Table 6.2 are provided by our templates. Our proposed method also yields competitive performance compared to alternative systems that utilize positive and negative examples of local features such as PBVLAD, CRN, and SRFS. Our method achieves higher results in R@N with an average of However, NetVLAD shows an improvement in performance, with average recalls of only 0.2 compared with our proposed method. On the other hand, our methods achieve better results in terms of computational cost, as the figure shows. Our method yields 0.0 better performance on average compared to PatchNetVLAD, despite a reduction in the number of local feature matchers.

In Table 6.3, our proposed method achieves the best performance, especially noticeable when considerable appearance differences are encountered in unseen environments, which are not normally employed for training procedures. The dataset from the social media image collection is used. Our method shows an improvement in the recall @N compared with NetVLAD. Note that all results shown in Table 6.3 are provided by our templates.

To analyze the effectiveness of our proposed method, we compared it with different types of feature selection. We evaluate the performance using negative features p^- only in the test phase. Before the geolocation process starts, the hard negative features in the query image compute the silimarties with the features in the reference dataset. The results in Table 6.4 show that the score is very low compared to the other selection processes. In addition, selecting all features to compare their similarity with all features in the reference dataset achieves better results than only negative selection. However, choosing both takes more time than choosing either positive or negative only. In general, selecting only positive features achieves better results in terms of recall and time. The figure shows the comparison between the three selection types in time computation.

Figure 6.13 shows the running time of our proposed method by applying asymptotic time complexity. We measure asymptotic time analysis using our method when selecting only relevant (positive) features and compare it when using all features. However, we have two algorithms with different running times. To determine which method is better, we measure asymptotic time complexity by focusing on the growth of the running time with increasing input size. As a result, figure 6.13 shows that our algorithm for selecting relevant features requires less time than our algorithm for using all features.

Qualitative Evaluation: Figure 6.5 demonstrates that the proposed method effectively retrieves the associated reference image. The Flicker image match example is particularly difficult because of the extreme viewpoint shift, occlusions, and things like automobiles and people on the street sides. Correct and incorrect findings are shown by green and red borders, respectively.

While CRN and PatchNetVLAD provide inaccurate location matches in Figure 6.6, our suggested technique successfully finds the corresponding reference image. CRN [55] and PatchNetVLAD [50] suffer from selecting the positive objects because the number of features in query images is huge.

6.4 conclusion

A contribution to visual place recognition has been made in this chapter, showing comprehensive quantitative and qualitative evaluations with leading state of the art visual place recognition methods. In particular, first, we show that the forcing method to learn or train negative examples does not improve performance. In contrast with the VPR methods, we train our method by removing the negative features from the query image prior to the geolocation process. Furthermore, we have demonstrated that place recognition performance for challenging real-world query images can be significantly improved by automatic prediction and suppression of spatially localized groups of confusing non-informative features in the query image. Confusing features are found by matching places spatially far on the map—a negative supervisory signal readily available in geo-tagged databases. We have also experimentally demonstrated that the method recognizes the place from hard environment changes by querying images taken from the social media platform. Finally, we improve on the method of the state of the art to better deal with efficient problems by predicting good features by taking advantage of GPS tags before localizing the palce. In this way, we discriminate images to find the place correctly while reducing the number of features.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Our research aimed to address the problem of how to recognize a place efficiently using only visual information in difficult environments. In reviewing research related to this problem, we identified particular gaps related to this research goal. We now list the main contributions made to our work.

- Mini-Batch VLAD for Visual Place Retrieval.

We investigated the visual place retrieval of an image query by utilizing a geotagged image dataset. For an efficient visual place retrieval system, the visual descriptors need to be discriminative. One of the high-dimensional descriptors that may be employed for accurate visual retrieval is the Vector of Locally Aggregated Descriptors (VLAD). VLAD describes the images by comparing an image's local feature descriptors to a previously computed codebook. Typically, a visual codebook is created by utilizing k-means to cluster the descriptors. However, determining sample distances in a large image collection is challenging due to the complexity of visual features, which is not straightforward. In order to develop an efficient image retrieval method with little computational expense, we suggested employing mini-batch k-means clustering to produce VLAD descriptors (MBVLAD). The proposed MBVLAD methodology beats state-of-the-art methods in retrieval accuracy.

- Efficient Visual Place Retrieval System Using Google Street View.

For the purpose of representing images in visual place retrieval, we proposed an aggregated binary descriptor with high-dimensional descriptors (MBVLAD). Local features allow it to successfully match local features across images; however, when huge databases are employed, the expense of extracting and comparing the local descriptors pairwise becomes a bottleneck. By aggregating binary local features, we may lower the cost of extracting, representing, and matching local visual descriptors, improving the efficiency of local features. Based on search accuracy (mAP) and search time (s), our experiments showed that our ORB-MBVLAD is much faster and has better search accuracy (mAP) than the other state-of-the-art approaches.

- Predicting relevant features using a hybrid feature for visual geolocation system.

Environmental change over a long-term life is one of the VPR system's most difficult challenges. The study of feature selection has received a lot of attention to solve this challenge. Finding a robust image representation that can handle the features of discrimination is tough. The majority of the features are handcrafted and have been shown to be extremely effective in visual place recognition. However, as hand-crafted representation exhibits superior recognition of a location in rotating perspectives but suffers from particular surroundings (i.e., trees, buildings, or mountains), it is challenging to choose what kinds of attributes should be employed to define places. With the fast advancement of deep learning networks, it is clear that learned features beat handcrafted features in place recognition tasks. In this thesis, we propose a hybrid feature as a robust image representation that can be used to distinguish relevant and non-relevant features. Additionally, a lot of features came from this phase, which reduced the efficiency of VPR when calculating similarity. In order to solve this problem, we used our proposed hybrid feature and the selecting features approach to predict useful features in a data-driven way. In conclusion, a thorough performance evaluation of various representation techniques was done. Compared to selecting features using a single method, our hybrid feature showed a significant improvement. We conducted extensive experiments and analysis that show quantitative and qualitative competitive results over the leading state of the art methods of visual place recognition.

- An Efficient Visual Place Recognition System by Predicting Unique Features.

We improved on the state-of-the-art method to better deal with this problem by proposing a method for automatically selecting such "mismatching features" and demonstrating that removing them from the query image can significantly improve the place recognition performance. To this end, we predicted good features by taking advantage of GPS tags prior to the geolocalization process. In this way, we discriminated against images to find the place correctly while reducing the number of features. In addition, our method provided increased accuracy with lower computational times. Extensive quantitative and qualitative experiments show competitive results as compared with the leading state of the art in visual place recognition.

7.2 Future Work

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

- Complex Scene Retrieval Using Semantics

In Chapters 3 and 4, we have taken into account reasonably well-defined retrieval situations where the ground-truth for a particular query is easily specified. In future research, however, we want to look into the complications that occur when the inquiry has a nuanced semantic meaning and the ground truth is only partly reliable. Take, for example, the query picture of a vehicle and a guy on a spooky, rain-soaked street. At first, it's not evident what the picture has to offer the user. Interactions from the user, including useful feedback or a bounding box, may aid in locating the target object(s). In order to answer complicated questions, current approaches use scene-graphs, which include first isolating each object and then parsing the result into a graph consisting of the identified items and their interactions [56, 139].

To further establish a link between the graph and the pictures, a graph inferencing technique is used. Therefore, the photos with the most similar spatial arrangements and linked components would be awarded the highest similarity ratings. In chapters 5 and 6, when a difficult scenario is shown, we'd want to look into several options for capturing its essence. Instead of recognizing each object independently and expressing the image as a graph, we aim to represent the image as a whole and detect frequently occurring visual concepts through a data-driven approach. This may involve chunks of objects, their coarse spatial configuration, and elements of the background. The visual parts that make up these concepts need to be tightly related to one another in order to reduce the amount of work required of the user. Users need a representation that can be broken apart so they can choose which parts of the image to show.

- Investigating Scene-Category-Aware Place Recognition

One of the difficulties with visual location recognition is scalability. In real time, we cannot do a similarity check on every image in the collection. By knowing the scene's semantic category, we may narrow the search field considerably. If we know that we are looking at a school, storefront, or building, for example, we may narrow the search to only include those images in the database that have a similar situation. However, semantic information may also be used to solve the issue of fluctuating appearance over time, which is another challenge in place recognition. In spite of cosmetic alterations, semantic data remains intact. When the seasons change, the physical look of a tree may alter drastically, yet the tree's underlying semantic category stays the same. Understanding this change, the trained models we provide in Chapters 5 and 6 of this study exclude fleeting visual components (such as trees, automobiles, and people strolling about) as irrelevant to the place recognition task. But if the important visual features have changed, like if a building's facade has been redone, our methods might not work unless there is a lot of overlap with how it looked before.

Bibliography

[1] Kashif Ahmad, Konstantin Pogorelov, Michael Riegler, Olga Ostroukhova, Pål Halvorsen, Nicola Conci, and Rozenn Dahyot. Automatic detection of passable roads after floods in remote sensed and social media data. *Signal Processing: Image Communication*, 74:110 – 118, 2019.

[2] Reem Aljuaidi and Rozenn Dahyot. Efficient visual place retrieval system using google street view. 2020.

[3] Reem Aljuaidi and Michael Manzke. An efficient visual place recognition system by predicting unique features. *Proceedings of the 5th International Conference on Computer Science and Software Engineering*, 2022.

[4] Reem Aljuaidi and Micheal Manzke. Predicting good features using a hybrid feature for visual geolocation system. In Xudong Jiang, Wenbing Tao, Deze Zeng, and Yi Xie, editors, *Fourteenth International Conference on Digital Image Processing (ICDIP 2022)*, volume 12342, page 123422D. International Society for Optics and Photonics, SPIE, 2022.

[5] Reem Aljuaidi, Jing Su, and Rozenn Dahyot. Mini-batch vlad for visual place retrieval. In *2019 30th Irish Signals and Systems Conference (ISSC)*, pages 1–6, June 2019.

[6] Shan An, Haogang Zhu, Dong Wei, Konstantinos A Tsintotas, and Antonios Gasteratos. Fast and incremental loop closure detection with deep features and proximity graphs. *Journal of Field Robotics*, 39(4):473–493, 2022.

[7] R. Arandjelovic and A. Zisserman. All about vlad. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, June 2013.

[8] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307. IEEE Computer Society, 2016.

[9] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 6328–6335. IEEE, 2015.

[10] Loukas Bampis, Angelos Amanatiadis, and Antonios Gasteratos. Fast loop-closure detection using visual-word-vectors from image sequences. *The International Journal of Robotics Research*, 37(1):62–82, 2018.

[11] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[12] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[13] Abdullah Bulbul and Rozenn Dahyot. Social media based 3d visual popularity. *Computers Graphics*, 63:28 – 36, 2017.

[14] César Cadena, Dorian Galvez-López, Juan D. Tardos, and José Neira. Robust place recognition with stereo sequences. *IEEE Transactions on Robotics*, 28(4):871–885, 2012.

[15] César Cadena, Dorian Gálvez-López, Fabio Ramos, Juan D. Tardós, and José Neira. Robust place recognition with stereo cameras. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5182–5189, 2010.

[16] Luis G Camara, Carl Gäbert, and Libor Přeučil. Highly robust visual place recognition through spatial matching of cnn features. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3748–3755. IEEE, 2020.

[17] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 700–707, 2013.

[18] David Caruso, Jakob Engel, and Daniel Cremers. Large-scale direct slam for omnidirectional cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148, 2015.

[19] Marvin Chancán, Luis Hernandez-Nunez, Ajay Narendra, Andrew B Barron, and Michael Milford. A hybrid compact neural architecture for visual place recognition. *IEEE Robotics and Automation Letters*, 5(2):993–1000, 2020.

[20] Zetao Chen, Adam Jacobson, Uğur M. Erdem, Michael E. Hasselmo, and Michael Milford. Multi-scale bio-inspired place recognition. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1895–1901, 2014.

[21] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230. IEEE, 2017.

[22] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509, 2014.

[23] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *I. J. Robotic Res.*, 27:647–665, 06 2008.

[24] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *I. J. Robotic Res.*, 30:1100–1123, 08 2011.

[25] Deyun Dai, Zonghai Chen, Jikai Wang, Peng Bao, and Hao Zhao. Robust visual place recognition based on context information. volume 52, pages 49–54. Elsevier, 2019.

[26] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[27] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[28] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009*

IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[29] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, and Ian Reid. Hm ˆ4: Hidden markov model with memory management for visual place recognition. *IEEE Robotics and Automation Letters*, 6(1):167–174, 2020.

[30] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

[31] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. *International Conference on Image and Video Retrieval*, 07 2009.

[32] Christian Eggert, Stefan Romberg, and Rainer Lienhart. Improving vlad: hierarchical coding and a refined local coordinate system. In *2014 IEEE international conference on image processing (ICIP)*, pages 3018–3022. IEEE, 2014.

[33] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *2012 IEEE International Conference on Robotics and Automation*, pages 1691–1696, 2012.

[34] Pablo Fernández Alcantarilla, Simon Stent, German Ros, Roberto Arroyo, and Riccardo Gherardi. Street-view change detection with deconvolutional networks. *Autonomous Robots*, 42, 10 2018.

[35] Torsten Fiolka, Jörg Stückler, Dominik A. Klein, Dirk Schulz, and Sven Behnke. Distinctive 3d surface entropy features for

place recognition. In *2013 European Conference on Mobile Robots*, pages 204–209, 2013.

[36] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[37] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[38] Amato G., Falchi F., and Vadicamo L. Aggregating binary local descriptors for image retrieval. volume 77, page 5385–5415. Kluwer Academic Publishers, Dordrecht ;, Stati Uniti d'America, 2018.

[39] Matthew Gadd, Daniele De Martini, and Paul Newman. Contrastive learning for unsupervised radar place recognition. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 344–349. IEEE, 2021.

[40] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[41] Sourav Garg and Michael Milford. Seqnet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robotics and Automation Letters*, 6(3):4305–4312, 2021.

[42] Sourav Garg, Niko Suenderhauf, and Michael Milford. Don't look back: Robustifying place categorization for viewpoint-and

condition-invariant place recognition. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3645–3652. IEEE, 2018.

[43] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 369–386. Springer, 2020.

[44] Toon Goedem©, Marnix Nuttin, Tinne Tuytelaars, and Luc Van Gool. Markerless computer vision based localization using automatically generated topological maps, 2004-05-01.

[45] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets (advances in neural information processing systems)(pp. 2672–2680). *Red Hook, NY Curran*, 2014.

[46] Petr Gronat, Guillaume Obozinski, Josef Sivic, and Tomas Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 907–914, 2013.

[47] Xiaobing Han, Yanfei Zhong, Liqin Cao, and Liangpei Zhang. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing*, 9(8), 2017.

[48] Peter Hansen and Brett Browning. Visual place recognition using hmm sequence matching. In *2014 IEEE/RSJ International*

Conference on Intelligent Robots and Systems, pages 4549–4555. IEEE, 2014.

[49] Qiang Hao, Rui Cai, Zhiwei Li, Lei Zhang, Yanwei Pang, and Feng Wu. 3d visual phrases for landmark recognition. In *Proc. of the 25th IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers, Inc., June 2012.

[50] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021.

[51] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[53] Ziyang Hong, Yvan Petillot, Andrew Wallace, and Sen Wang. Radarslam: A robust simultaneous localization and mapping system for all weather conditions. *The International Journal of Robotics Research*, 41(5):519–542, 2022.

[54] Forrest N. Iandola, Matthew W. Moskewicz, K. Ashraf, Song Han, W. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *ArXiv*, abs/1602.07360, 2017.

[55] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[56] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[57] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, June 2010.

[58] Leonard Kaufmann. Clustering by means of medoids. In *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987*, pages 405–416, 1987.

[59] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Convolutional networks for real-time 6-dof camera relocalization. *CoRR*, abs/1505.07427, 2015.

[60] Ahmad Khaliq, Shoaib Ehsan, Zetao Chen, Michael Milford, and Klaus McDonald-Maier. A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. *IEEE transactions on robotics*, 36(2):561–569, 2019.

[61] Giseop Kim, Sunwook Choi, and Ayoung Kim. Scan context++: Structural place recognition robust to rotation and lateral vari-

ations in urban environments. *IEEE Transactions on Robotics*, 2021.

[62] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018.

[63] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1170–1178, Washington, DC, USA, 2015. IEEE Computer Society.

[64] Yong Nyeon Kim, Dong Wook Ko, and Il Hong Suh. Visual navigation using place recognition with visual line words. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 676–676, 2014.

[65] Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 748–761, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[66] Xin Kong, Xuemeng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Feng Wen. Semantic graph based place recognition for 3d point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8216–8223. IEEE, 2020.

[67] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.

[68] Vladimir A. Krylov, Eamonn Kenny, and Rozenn Dahyot. Automatic discovery and geotagging of objects from street view imagery. *Remote Sensing*, 10(5), 2018.

[69] Gerhard Kurz, Matthias Holoch, and Peter Biber. Geometry-based graph pruning for lifelong slam. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3313–3320. IEEE, 2021.

[70] Hyo Jong Lee, Ihsan Ullah, Weiguo Wan, Yongbin Gao, and Zhijun Fang. Real-time vehicle make and model recognition with the residual squeezenet architecture. *Sensors*, 19(5), 2019.

[71] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. Rinet: Efficient 3d lidar-based place recognition using rotation invariant neural network. *IEEE Robotics and Automation Letters*, 7(2):4321–4328, 2022.

[72] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, ECCV'10, pages 791–804, Berlin, Heidelberg, 2010. Springer-Verlag.

[73] Yunpeng Li, Noah Snavely, Daniel P Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. *Large-Scale Visual Geo-Localization*, pages 147–163, 2016.

[74] Hyon Lim, Sudipta N Sinha, Michael F Cohen, and Matthew Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1043–1050. IEEE, 2012.

[75] Yang Liu and Hong Zhang. Towards improving the efficiency of sequence-based slam. In *2013 IEEE International Conference on Mechatronics and Automation*, pages 1261–1266. IEEE, 2013.

[76] Zhe Liu, Chuanzhe Suo, Shunbo Zhou, Fan Xu, Huanshu Wei, Wen Chen, Hesheng Wang, Xinwu Liang, and Yun-Hui Liu. Seqlpd: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1218–1223. IEEE, 2019.

[77] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[78] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David Cox, Peter Corke, and Michael J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[79] Huimin Lu, Kaihong Huang, Dan Xiong, Xun Li, and Zhiqiang Zheng. A robust place recognition algorithm based on om-

nidirectional vision for mobile robots. In Xiaoping Chen, Peter Stone, Luis Enrique Sucar, and Tijn van der Zant, editors, *RoboCup 2012: Robot Soccer World Cup XVI*, pages 286–297, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[80] Will Maddern, Michael Milford, and Gordon Wyeth. Catslam: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research*, 31(4):429–451, 2012.

[81] Will Maddern, Michael Milford, and Gordon Wyeth. Towards persistent indoor appearance-based localization, mapping and navigation using cat-graph. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4224–4230, 2012.

[82] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[83] Nathaniel Merrill and Guoquan Huang. Calc2. 0: Combining appearance, semantic and geometric information for robust and efficient visual loop closure. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4554–4561. IEEE, 2019.

[84] Michael J Milford, Ian Turner, and Peter Corke. Long exposure localization in darkness using consumer cameras. In *2013 IEEE International Conference on Robotics and Automation*, pages 3755–3761. IEEE, 2013.

[85] Timothy L Molloy, Tobias Fischer, Michael Milford, and Girish N Nair. Intelligent reference curation for visual place recognition

via bayesian selective fusion. *IEEE Robotics and Automation Letters*, 6(2):588–595, 2020.

[86] Hiroshi Morioka, Sangkyu Yi, and Osamu Hasegawa. Vision-based mobile robot's slam and navigation in crowded environments. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3998–4005, 2011.

[87] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[88] A. C. Murillo and J. Kosecka. Experiments in place recognition using gist panoramas. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2196–2203, 2009.

[89] Ana Murillo, Gautam Singh, J. Kosecka, and Josechu Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29:146–160, 02 2013.

[90] P. Newman, D. Cole, and K. Ho. Outdoor slam using visual appearance and laser ranging. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1180–1187, 2006.

[91] Dat Tien Nguyen, Tuyen Danh Pham, Na Rae Baek, and Kang Ryoung Park. Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors. *Sensors*, 18(3):699, 2018.

[92] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168, 2006.

[93] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006.

[94] Michał R Nowicki, Jan Wietrzykowski, and Piotr Skrzypczyński. Real-time visual place recognition for personal localization on a mobile device. *Wireless Personal Communications*, 97(1):213–244, 2017.

[95] Anicetus Odo, Stephen McKenna, David Flynn, and Jan Vorstius. Towards the automatic visual monitoring of electricity pylons from aerial images. In Giovanni Maria Farinella, Petia Radeva, and Jose Braz, editors, *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 5, pages 566–573, Portugal, February 2020. VISAPP. VISAPP 2020 : 15th International Conference on Computer Vision Theory and Applications ; Conference date: 27-02-2020 Through 29-02-2020.

[96] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 05 2001.

[97] Valerio Paolicelli, Antonio Tavera, Carlo Masone, Gabriele Berton, and Barbara Caputo. Learning semantics for visual

place recognition through multi-scale attention. In *International Conference on Image Analysis and Processing*, pages 454–466. Springer, 2022.

[98] Jiman Park, Jihang Kim, and Byungyun Yang. Spatializing an artist-resident community area at a building-level: A case study of garosu-gil, south korea. *Sustainability*, 12(15), 2020.

[99] Luis Payá, Lorenzo Fernández, Arturo Gil, and Óscar Reinoso. Map building and monte carlo localization using global appearance of omnidirectional images. *Sensors (Basel, Switzerland)*, 10:11468–97, 12 2010.

[100] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[101] Edward Pepperell, Peter I Corke, and Michael J Milford. All-environment visual place recognition with smart. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 1612–1618. IEEE, 2014.

[102] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. IEEE Computer Society, 2007.

[103] Nathan Piasco, Désiré Sidibé, Valérie Gouet-Brunet, and Cédric Demonceaux. Improving image description with auxiliary modality for visual localization in challenging conditions. *International Journal of Computer Vision*, 129(1):185–202, 2021.

[104] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[105] A. Rituerto, A. C. Murillo, and J. J. Guerrero. Semantic labeling for indoor topological mapping using a wearable catadioptric system. *Robotics Auton. Syst.*, 62:685–695, 2014.

[106] Duncan P. Robertson and Roberto Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.

[107] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

[108] Torsten Sattler, Michal Havlena, Filip Radenović, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2102–2110, 2015.

[109] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1582–1590, 2016.

[110] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localiza-

tion? In *2017 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE/CVF, IEEE, July 2017.

[111] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007.

[112] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1177–1178, New York, NY, USA, 2010. ACM.

[113] Albert Shaw, Daniel Hunter, Forrest Landola, and Sammy Sidhu. Squeezenas: Fast neural architecture search for faster semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[114] Sayem Mohammad Siam and Hong Zhang. Fast-seqslam: A fast appearance based place recognition algorithm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5702–5708. IEEE, 2017.

[115] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[116] Gautam Singh. Visual loop closing using gist descriptors in manhattan world. In *in Omnidirectional Robot Vision workshop, held with IEEE ICRA*, 2010.

[117] Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE In-*

ternational Conference on Computer Vision, pages 1470–1477 vol.2, 2003.

[118] Josef Sivic, Bryan C. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 1:370–377 Vol. 1, 2005.

[119] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society, 2003.

[120] Thomas Stone, Dario Differt, Michael Milford, and Barbara Webb. Skyline-based localisation for aggressively manoeuvring robots using uv sensors and spherical harmonics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5615–5622. IEEE, 2016.

[121] Ting Sun, Ming Liu, Haoyang Ye, and Dit-Yan Yeung. Point-cloud-based place recognition using cnn feature extraction. *IEEE Sensors Journal*, 19(24):12175–12186, 2019.

[122] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, 2015.

[123] Ben Talbot, Sourav Garg, and Michael Milford. Openseqslam2. 0: An open source toolbox for visual place recognition under changing conditions. In *2018 IEEE/RSJ international confer-*

ence on intelligent robots and systems (IROS), pages 7758–7765. IEEE, 2018.

[124] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: Aggregation across single and multiple images. *International Journal of Computer Vision*, 116:247–261, 2015.

[125] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.

[126] Konstantinos A Tsintotas, Loukas Bampis, and Antonios Gasteratos. Doseqslam: Dynamic on-line sequence based loop closure detection algorithm for slam. In *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE, 2018.

[127] Konstantinos A Tsintotas, Panagiotis Giannis, Loukas Bampis, and Antonios Gasteratos. Appearance-based loop closure detection with scale-restrictive visual features. In *International Conference on Computer Vision Systems*, pages 75–87. Springer, 2019.

[128] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1023–1029 vol.2, 2000.

[129] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4470–4479, 2018.

[130] Dominik Van Opdenbosch, Georg Schroth, Robert Huitl, Sebastian Hilsenbeck, Adrian Garcea, and Eckehard Steinbach. Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In *2014 IEEE international conference on image processing (ICIP)*, pages 2804–2808. IEEE, 2014.

[131] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[132] Junqiu Wang and Yasushi Yagi. Robust location recognition based on efficient feature integration. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 97–101, 2012.

[133] Min-Liang Wang and Huei-Yung Lin. A hull census transform for scene change detection and recognition towards topological map building. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 548–553, 2010.

[134] Christian Weiss, Andreas Masselli, Hashem Tamimi, and Andreas Zell. Fast outdoor robot localization using integral invariants. 03 2007.

[135] Tobias Weyand, André Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2572–2581, 2020.

[136] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[137] Khin Yadanar Win, Noppadol Maneerat, Kazuhiko Hamamoto, and Syna Sreng. Hybrid learning of hand-crafted and deep-activated features using particle swarm optimization and optimized support vector machine for tuberculosis screening. *Applied Sciences*, 10(17):5749, 2020.

[138] Yifan Xia, Jie Li, Lin Qi, and Hao Fan. Loop closure detection for visual slam using pcanet features. In *2016 international joint conference on neural networks (IJCNN)*, pages 2274–2281. IEEE, 2016.

[139] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[140] Peng Yin, Rangaprasad Arun Srivatsan, Yin Chen, Xueqian Li, Hongda Zhang, Lingyun Xu, Lu Li, Zhenzhong Jia, Jianmin Ji, and Yuqing He. Mrs-vpr: a multi-resolution sampling based global visual place recognition method. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7137–7142. IEEE, 2019.

[141] Peng Yin, Fuying Wang, Anton Egorov, Jiafan Hou, Zhenzhong Jia, and Jianda Han. Fast sequence-matching enhanced viewpoint-invariant 3-d place recognition. *IEEE Transactions on Industrial Electronics*, 69(2):2127–2135, 2021.

[142] Peng Yin, Fuying Wang, Anton Egorov, Jiafan Hou, Ji Zhang, and Howie Choset. Seqspherevlad: Sequence matching enhanced orientation-invariant place recognition. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5024–5029. IEEE, 2020.

[143] Peng Yin, Lingyun Xu, Ziyue Feng, Anton Egorov, and Bing Li. Pse-match: A viewpoint-free place recognition method with parallel semantic embedding. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[144] Peng Yin, Lingyun Xu, Ji Zhang, and Howie Choset. Fusionvlad: A multi-view deep fusion networks for viewpoint-free 3d place recognition. *Ieee Robotics and Automation Letters*, 6(2):2304–2310, 2021.

[145] Mubariz Zaffar, Shoaib Ehsan, Michael Milford, and Klaus McDonald-Maier. Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. *IEEE Robotics and Automation Letters*, 5(2):1835–1842, 2020.

[146] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 255–268, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[147] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014.

[148] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, page 33–40, USA, 2006. IEEE Computer Society.

[149] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019.

[150] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021.

[151] Yu Zhang, Chao Zhu, Stephane Bres, and Liming Chen. Encoding local binary descriptors by bag-of-features with hamming distance for visual object categorization. In *European Conference on Information Retrieval*, pages 630–641. Springer, 2013.

[152] Lanyue Zhi, Zhifeng Xiao, Yonggang Qiang, and Linjun Qian. Street-level image localization based on building-aware features via patch-region retrieval under metropolitan-scale. *Remote Sensing*, 13(23), 2021.

[153] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Appendices

Appendix A

Image Representation Methods Using Three Dimentional Information for Visual Place Recognition

In chapter 2, we explore the image representation methods for visual place retrieval and recognition. One of the representation methods that can be used for visual place retrieval is three-dimensional information.

A.1 Image Representation Methods Using Three Dimentional Information for VPR

The visual domain may be used to describe locations using a 2D model (instead of creating a geometric-model), and this can be augmented with metric data. Therefore, a two-dimensional (2D) image that includes metric data may be treated as if it were three-dimensional (3D). It is possible to infer distances in meters using stereo cameras.

Structure-from-motion algorithms, such as those used in MonoSLAM [27], LSD-SLAM [18], and ORB-SLAM [87], allow monocular cameras to infer metric information. In order to boost the efficiency of localization strategies based on place recognition, some publications in the literature have turned to 3D (three-dimensional) data. Fast Appearance-Based Mapping (FAB-MAP) is improved upon by include the 3D spatial distribution of visual words [23]. Similarly, [14] uses visual words and 3D information from stereo sequences to accomplish accurate place recognition. Extracting robust 3D PIRF (Position Invariant Robust Feature) points from sequential images and odometry, Morioka et al. [86] offer a SLAM navigation system that is successful even in congested surroundings. The authors of the study [35] offer a variation of SURE, an interest point detector and descriptor for 3D point clouds and depth images, and show how it may be used to identify semantically unique locations within buildings. In addition, they showed that a bag-of-words technique works very well for recognizing locations utilizing SURE features. In order to increase the regularity and dependability of appearance-based loop closure, the authors of the paper [80] describe a new system called CAT-SLAM (Continuous Appearance based Trajectory SLAM), which combines sequential appearance-based place recognition with local metric pose filtering. In [81], we offer CAT-Graph, a version of CAT-SLAM that, in addition to merging visual appearance and local odometry data, also fuses data from many trips to the same site into a topological graph-based description of interior settings. It illustrates that, with limited resources (computing time and memory), loop closure detection in a dense urban setting may achieve performance that is three times better than FAB MAP at 100 accuracy. According to Cadena et al. [15] presents a stereo vision-based framework for location identification that uses a bag-of-words model to find potential loop closures and a Conditional Random Fields-Matching (CRF-Matching) technique to confirm them. When compared to methods that rely just on epipolar geometry, this matching method's utilization of 3D information offered by the stereo images makes it far more efficient.

The paper [14] provides a method for recognizing places in camera-based SLAM systems. It takes into account both the visual features and the geometric properties of places of interest in the images. Loop-closing hypotheses are created rapidly using an appearance method based on the bag-of-words approach. Experiments with both indoor and outdoor data demonstrate that the suggested approach achieves high recall while maintaining high accuracy (i.e., no false positives) (fewer false negatives). Sensor information from supplementary devices, including RGB-D cameras, is used by a wide variety of other systems [33].

Appendix B

Place Match Techniques in Visual Place Retrieval and Recognition

Visual place recognition can be divided into two categories: 1) methods based on single image matching; 2) methods based on sequence matching. In this thesis, we only focus on the single image matching technique. In the following, we provide an overview of methods based on single image and sequence matching.

B.1 Matching Methods based on single image

It has been common practice to consider a place as a single image when doing visual place recognition. The foundation of the method is the compilation of images captured offline by a moving machine. It is then possible to obtain the one that is most like the one being used at the moment. It's possible to attribute the same origin to disparate locales if their similarities are great enough. Visual similarity retrieval

(VPR) requires that for each image in the test set, a matching image be found in the reference set. Geo-locating an image involves matching its coordinates to those of an image of the same location that was pulled from a database as a starting place (map). Matching the visual features of the current scene image to those of training images from a database is a key step in many visual localization approaches based on place recognition [78].

For visual localization, FAB-MAP [23] stands out as a pioneering picture matching approach. The paper suggests using a bag-of-words image retrieval approach to find a image that most closely resembles the present scene's look. Training involves calculating the uniqueness of each word in a bag-of-words model that employs SIFT or SURF features for image description. A Chow Liu tree, which is the maximum-weight spanning tree of a directed network of co occurrences of visual words, is calculated from a set of training data to approximatively represent the probabilities of visual words. In order to solve the perceptual aliasing issue, FAB-MAP takes into account not only the similarity between two places in terms of the number of visual words they share, but also the rarity of those words. However, when doing large-scale appearance-based localization, Knopp et al. [65] only take into account matches to individual images in the database, rather than the linear combination of bag-of-feature vectors. Matching a single image is a simple and fast method for recognizing a certain location. Place identification with a single image, however, may be impacted by changes in lighting and moving objects when robotic systems work in bigger, uncontrolled areas and for longer periods of time (e.g. cars or pedestrians).

B.2 Image Matching Methods Based on Sequence Matching

The visual appearance of each location was assumed to remain constant during the length of the trial, a simplification that was commonly made implicitly in early place recognition systems. This assumption, however, has rapidly been shown to be incorrect as robotic systems work in ever-larger and longer-lasting uncontrolled settings. The relative topological structure of an environment becomes increasingly essential and appearance-based location matching becomes less trustworthy when the look of an environment is changing. Sequences of images may be utilized to match locations in spite of variations in lighting, weather, or visibility [85], as opposed to estimating the similarity between a single position in two images. Even if the lighting and objects (cars and trees) in sequence A have changed over the course of the two weeks between sequences A and A', matching based on sequence is still able to correctly identify the location.

SeqSLAM (Sequence Simultaneous Localization and Mapping) [85] is a more recent approach that incorporates the concept of matching locations by focusing on sequences rather than individual images. The first step in the SeqSLAM procedure is to create a matrix that compares the training picture sequence to the local query (testing) image sequence. Without extracting keypoints from images, similarity is measured by summing up the differences between contrast-enhanced versions of low-resolution images. The place recognition score is the greatest normalized sum of the similarity scores along the prescribed constant velocity pathways (alignments between the query sequence and database sequence images) in the matrix. Using this sequence matching strategy dramatically enhances location recognition reliability. As long as the right position is more similar than a wrong site, the sequence-based technique can work consistently under these circumstances. This is because it does not rely on the image comparison phase to ensure 100 accuracy. sequence filter can determine the route if it occurs often enough [84].

Appendix C

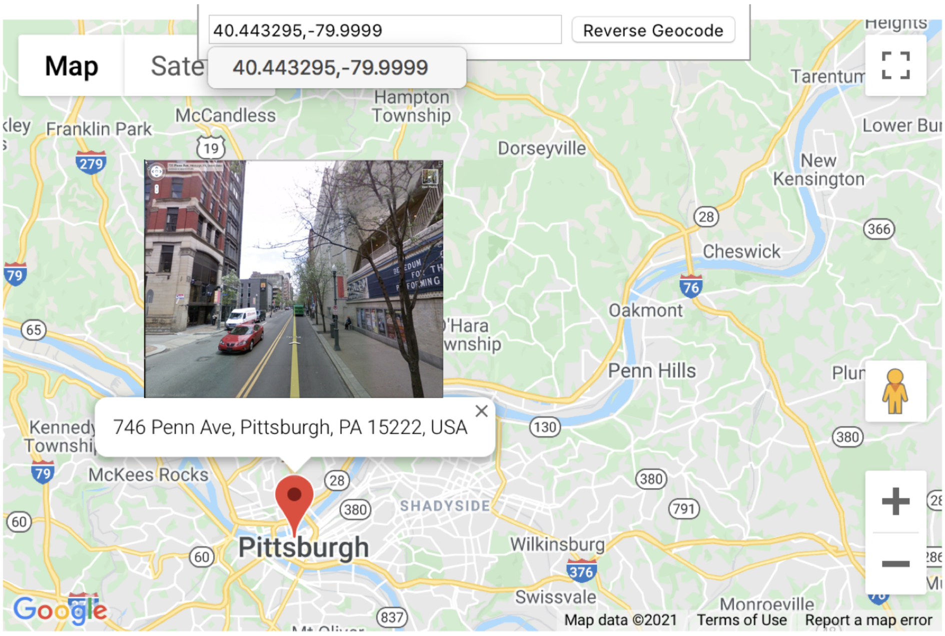
Geo-tagged image collection

In chapters 5 and 6, we need a geo-tagged dataset that contains images with geographical tags for training features. There is no dataset that covers the same geographic area as Google Street View dataset zamir2014image. So, we collect our images from free social media platforms. In the following, we explain the collection in more detail.

C.1 Images from Flicker API

Our geotagged images contains 720 images that can be used for training, and 100 images for testing. The images can be used for many computer vision tasks such as place retrieval, visual place recognition, and image classification. The images include street-level images taken from multiple perspectives, at different but nearby locations. Figure illustrates a set of images for the same location but different perspectives views.

Our collected dataset covers cityscape views, major buildings, and outdoors landmarks in Pittsburgh, PA. Figure presents an example of our collected dataset. The Flickr API allows images to be selected by tag. For example, we can search by objects, places and buildings such as lamp-post, pizza, church, etc. For our dataset, we



40.443295,-79.9999

Reverse Geocode

40.443295,-79.9999

746 Penn Ave, Pittsburgh, PA 15222, USA

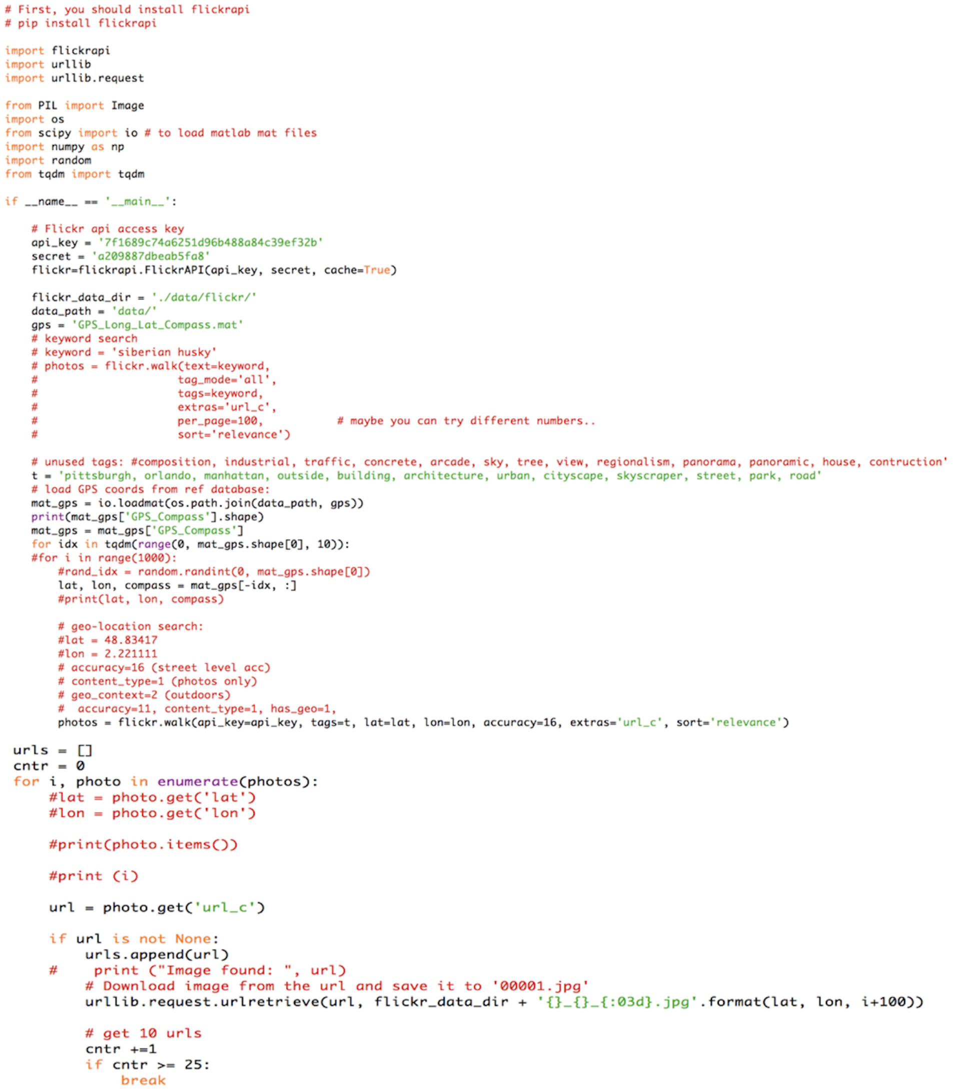
Figure C.1: An example of using Google Map Platform to verify the GPS tags.

searched by GPS tag, using geo-location (latitude and longitude) from the Google Street View Dataset provided by Zamir and Shah [147].

C.2 Implementation details

To collect geo-tagged images from Flicker API, we should first request for API key access from the website. For collecting the photos, Python 3.5 is used. The Flicker API should install by $pip\ install\ flickrapi$. The python code for searching and collecting the photos shows in Figure C.2. At the end, all images collected in one folder. Each image has a tag like image number-longitude-latitude.jpg. We also verify any coordinates using Google Map Platform. [1], see Figure C.1.

[1] https://developers.google.com/maps/documentation/javascript/examples/geocoding-reverse



```
# First, you should install flickrapi
# pip install flickrapi

import flickrapi
import urllib
import urllib.request

from PIL import Image
import os
from scipy import io # to load matlab mat files
import numpy as np
import random
from tqdm import tqdm

if __name__ == '__main__':

    # Flickr api access key
    api_key = '7f1689c74a6251d96b488a84c39ef32b'
    secret = 'a209887dbeab5fa8'
    flickr=flickrapi.FlickrAPI(api_key, secret, cache=True)

    flickr_data_dir = './data/flickr/'
    data_path = 'data/'
    gps = 'GPS_Long_Lat_Compass.mat'
    # keyword search
    # keyword = 'siberian husky'
    # photos = flickr.walk(text=keyword,
    #                      tag_mode='all',
    #                      tags=keyword,
    #                      extras='url_c',
    #                      per_page=100,           # maybe you can try different numbers..
    #                      sort='relevance')

    # unused tags: #composition, industrial, traffic, concrete, arcade, sky, tree, view, regionalism, panorama, panoramic, house, contruction'
    t = 'pittsburgh, orlando, manhattan, outside, building, architecture, urban, cityscape, skyscraper, street, park, road'
    # load GPS coords from ref database:
    mat_gps = io.loadmat(os.path.join(data_path, gps))
    print(mat_gps['GPS_Compass'].shape)
    mat_gps = mat_gps['GPS_Compass']
    for idx in tqdm(range(0, mat_gps.shape[0], 10)):
    #for i in range(1000):
        #rand_idx = random.randint(0, mat_gps.shape[0])
        lat, lon, compass = mat_gps[-idx, :]
        #print(lat, lon, compass)

        # geo-location search:
        #lat = 48.83417
        #lon = 2.221111
        # accuracy=16 (street level acc)
        # content_type=1 (photos only)
        # geo_context=2 (outdoors)
        #  accuracy=11, content_type=1, has_geo=1,
        photos = flickr.walk(api_key=api_key, tags=t, lat=lat, lon=lon, accuracy=16, extras='url_c', sort='relevance')

urls = []
cntr = 0
for i, photo in enumerate(photos):
    #lat = photo.get('lat')
    #lon = photo.get('lon')

    #print(photo.items())

    #print (i)

    url = photo.get('url_c')

    if url is not None:
        urls.append(url)
    #    print ("Image found: ", url)
        # Download image from the url and save it to '00001.jpg'
        urllib.request.urlretrieve(url, flickr_data_dir + '{}_{}_{:03d}.jpg'.format(lat, lon, i+100))

        # get 10 urls
        cntr +=1
        if cntr >= 25:
            break
```

Figure C.2: A screen shot of the python code used for collecting the images from Flicker API.

Appendix D

Fast Nearest Neighbor Search

In this dissertation, we have concentrated on representing an image using aggregated approaches, where visual similarity is measured by the distance in the embedding space. Computing the distance between the query and all images in the database is computationally intensive and frequently impractical. Numerous efforts have been made to improve scalability via the use of tree-search, hashing, and quantization, among other techniques. Unless otherwise noted, the strategies discussed in this section may be utilized in conjunction with the methods we suggest.

D.1 Indexing using Vocabulary Tree

When applied to an inverted index, the vocabulary tree [93] allows for a fast closest neighbor search for representations of a bag of visual words [119]. Each node in a vocabulary tree represents a distinct word in a visual language, making it a hierarchical quantizer. On the other hand, the inverted index is a data structure that associates visual words with the image indexes that include those words. It takes O(KL), where K is the number of centroids per level and L is the tree depth, to assign a query image's locally derived features to a visual word through the vocabulary tree. For each visual word in the query image, the inverted index increases the number of co-occurring words for the list of images returned. This aids in the rapid locate similar images.

D.2 KD-Tree

Via a vector representation, a KD-Tree [37] locates near-neighbors using a tree search. At first, a tree is built by recursively dividing the database in half along one of the k feature dimensions at the median, and continuing to do so until the partitioning conditions (the number of data points or the greatest distance between the points at the node) are met. It's like taking the concept of a binary tree and expanding it to k dimensions. The tree is explored in a recursive manner using depth-first search for a particular query image. In the leaf pile, a leaf will discover its closest neighbors when it is picked up. If a new neighbor is discovered, the distance between the neighbors is recalculated, and a new bounding box is drawn. If the bounding box does not cover a portion of the sub tree, then that portion of the sub tree is skipped.