PH.D THESIS

A Framework for Automated Landmark Recognition in Community Contributed Image Corpora

by

Mark Hughes, B.Sc. (Hons)

School of Computing

Supervisor:

Dr. Gareth Jones

September 16, 2011

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D in Computer Science is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____ ID No.: _____
Mark Hughes

Date: _____

Acknowledgements

Firstly I would like to thank my supervisor Dr. Gareth Jones for his guidance and support over the last four years. I would also like to express sincere thanks to Prof. Noel O'Connor for all his advice and encouragement during the duration of the work described in this thesis. Thanks must also go to Dr. Andrew Salway, for the many hours that he spent discussing possible research avenues and providing valuable insight in the PhD process. Thanks to Prof. Alan Smeaton for giving me the opportunity to join the Centre for Digital Processing and partake in this research.

A special thank you to Dr Ciaran MacAnBhaird for his invaluable advice and time consuming help. Thanks are also due to Dr. Neil O'Hare for his thoughts and valuable assistance. Thanks to Daragh Byrne for developing the iPhone application that is described in this work. Many days of developing and debugging were much appreciated.

Thank you to the Tripod FP6 STREP Project for funding this research. A big thanks also goes to all the people associated with the Tripod project. In particular the project coordinators Mark and Ross, and to Phil, Eduardo, Alistair and Christian for all their advice and technical knowledge, that was kindly shared.

I am also grateful to all the current and past members of the CDVP and Clarity research centres, that have contributed research ideas, advice and materials over the years.

I thank my parents, Brian and Pauline, and sisters Olivia, Lorraine and Eimear for all the years of support.

Last but not least I would like to thank Sanita, for all your love and encouragement throughout. Liels paldies.

Abstract

Any large library of information requires efficient ways to organise it and methods that allow people to access information efficiently and collections of digital images are no exception. Automatically creating high-level semantic tags based on image content is difficult, if not impossible to achieve accurately. In this thesis a framework is presented that allows for the automatic creation of rich and accurate tags for images with landmarks as the main object. This framework uses state of the art computer vision techniques fused with the wide range of contextual information that is available with community contributed imagery.

Images are organised into clusters based on image content and spatial data associated with each image. Based on these clusters different types of classifiers are* trained to recognise landmarks contained within the images in each cluster. A novel hybrid approach is proposed combining these classifiers with an hierarchical matching approach to allow near real-time classification and captioning of images containing landmarks.

Abbreviations and Acronyms

3G Third Generation

4G Fourth Generation

API Application Programming Interface

BBF Best Bin First

BOW Bag of Words

CBIR Content-Based Image Retrieval

DHC Divisive Hierarchical Clustering

DOG Difference of Gaussian

EHD Edge Histogram Descriptor

GLCM Grey Level Co-Occurrence Matrix

GPS Global Positioning System

HKM Hierarchical K-Means

HOG Histogram of Gradients

HSV Hue, Saturation and Value

k-NN k-Nearest Neighbour

LoG Laplacian of Gaussian

LSH Locality Sensitive Hashing

MOPS Multi-Scale Oriented Patches

NN Nearest Neighbour

MPEG Moving Pictures Experts Group

POI Point of Interest

QBIC Query by Image Content

RANSAC Random Sample Consensus

RBF Radial Basis Function

REST Representational State Transfer

SIFT Scale-Invariant Feature Transform

SURF Speeded Up Robust Features

SCD Scalable Colour Descriptor

SVM Support Vector Machines

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Over recent years the fast paced growth of technology has led to innovations that allow people to capture a digital image, automatically associate valuable metadata with that image, upload it to an on-line repository, and allow that image to be shared and viewed around the world. As a consequence, there has been significant growth in the amount of digital imagery that is being stored on-line. Approaches are sought that allow for the efficient organisation and retrieval of these images in a timely and precise manner. The work presented in this thesis, aims to address some of the problems associated with recognising and retrieving images relevant to a user's query in large scale image corpora.

In this chapter an introduction to the subject of the thesis is provided, along with the motivations behind the work and the main research objectives. Firstly, an overview to the research problem addressed in this thesis is provided, together with a brief description of a proposed solution to this problem. The next section describes the motivating factors behind this research, and describes why the solutions currently used are inefficient. Two hypotheses are then proposed and outlined followed by the main research objectives of this work. This chapter concludes with an outline of the thesis and brief description of each chapter.

1.1 Overview

The main aim of this work is to recognise images that contain a landmark as the main subject of a photograph, and to recognise the actual landmark depicted. A landmark is defined as a unique man-made object or geographical feature in a specific location, that is generally considered unique from other objects in the region. For example, an object that a tourist might associate with a region, and possibly photograph, could be considered a landmark. Several types of man-made objects could be considered landmarks, such as bridges, unique buildings, churches, fountains and statues among many others. The main aim of this work is to provide a framework that allows for the recognition of these landmarks in a memory efficient, automated manner and acceptable timeframe.

An acceptable timeframe is defined as near real-time recognition or more specifically a timeframe that is tolerable for users in an interactive application. It is envisaged that this framework could be successfully integrated into a mobile image recognition platform (described in Appendix A) for use by large numbers of people. Therefore it is important to perform these tasks within a set timeframe that is considered tolerable to users. In work carried out by Hoxmeier [Hoxmeier et al., 2000] it was determined that an upper threshold for the tolerable waiting time for users in a browser environment for complex tasks was approximately 12 seconds. It is assumed that with a mobile based device users would be a little more tolerant due to additional complexity, and issues with internet connection speeds.

It is very difficult to classify high level semantics, such as names of depicted landmarks, from an image using content alone in an unconstrained environment. One approach to automatic landmark classification is to harvest a large collection of annotated landmark images and match input images to this collection, based on context and content features extracted from the input image

[Qingji et al., 2008],[Rahmani et al., 2008],[Zhang and Kosecka, 2007]. However, many of these approaches can lead to inaccurate results, and can not be achieved in suitable time frames. Additionally, some of these approaches can also be memory inefficient and restrictions can apply to the maximum size of a training corpus.

The majority of these approaches are based upon a type of image feature that describe small regions within an image. These localised image features are commonly called interest points. Image and object matching using interest point features has been shown to work well even in large-scale image databases containing many different images [Konolige et al., 2009]. These localised features, tend to be more discriminate and less sensitive to occlusion than traditional global based features. The majority of landmark recognition algorithms that have been suggested to date, are based upon the matching of these interest point features.

Brute force matching between keypoints is computationally expensive, and with very large image databases will be computationally infeasible. Although it depends on image content and size, each commonly used interest point detection method will generate on average up to 1000 keypoints from an image [Lowe, 2004]. This presents a considerable challenge in terms of matching two images using their interest points and means significant computational overhead. To put it into perspective, to compare one image to all images in a 1000 image dataset using the Scale Invariant Feature Transform (SIFT) algorithm [Lowe, 2004], would require 128 million comparisons to be made (1000 images $\times$ 1000 keypoints $\times$ 128 values per keypoint vector). To compare one image against a dataset of 100,000 images, would require over 12 trillion comparisons to be made and this number would grow considerably as the size of the dataset grew. Clearly, this type of brute force matching could not be done in real time with large scale image collections, which is required in this work, due to the sheer number of images which could be uploaded and would be required to be processed daily.

In order to achieve the objectives outlined in this chapter, techniques are required which will filter the amount of keypoints that need to be compared, or alternatively techniques that do not match keypoint by keypoint individually, in order to be able to do this matching in real time. In this thesis, a framework is hypothesised, that is based on existing computer vision techniques. These are merged with a number of unique ways to organise data, fusing different forms of semantic contextual data with image content features. The aim is to improve upon the current state of the art image matching methods (with regards to classification accuracy, speed and memory efficiency), most of which do not scale well with regards to memory and processing time constraints when using very large datasets. A large scale dataset is defined as a corpus consisting of tens of thousands and possibly even millions of images. The framework described in this thesis is then implemented and applied to an image collection, created from community contributed data.

Viewpoint clustering involves taking many images of the same landmark from a relatively similar viewpoint and clustering them together to create groups of images that are visually similar. This framework is based on the concept of viewpoint clustering, which allows for the efficient and accurate classification of landmarks in a memory efficient manner, using a large scale training dataset. Once these clusters are created and assigned spatial data, they should allow for the classification of test images based on machine learning classification models using a machine learning algorithm such as Support Vector Machines (SVMs).

In this work, a large number of machine learning classification models are trained to recognise large landmarks within an image, with one class of training features in each classification model representing a single landmark from a certain viewpoint. The main advantages of utilising machine learning techniques such as SVMs for image classification is that they are robust and accurate, and allow for quick classification when used in conjunction with appropriate features. In

the past, SVM classification models have been used with local image features to classify scenes into high level semantics [Bosch et al., 2008], along with scene localisation [Ayers and Boutell, 2007] and have been shown to work efficiently and accurately. One major drawback with using this technique however, is that a relatively large number of positive examples are needed to classify an image correctly, which might not be available for all landmarks in a dataset. In this work, this issue is addressed by the proposal of a hybrid classification approach that avoids expensive point to point matching and allows for the classification of landmark images in situations where there is insufficient data to train an accurate SVM model. This hybrid approach is based on combining spatially organised SVM models with a divisive hierarchical classification algorithm [Lamrous and Taileb, 2006] that selects candidate matches from a large dataset using computationally inexpensive methods, before confirming matches using more sophisticated image matching schemes.

1.2 Motivation

With the arrival of consumer digital cameras and the continuous reduction in the cost of these devices, an average consumer now has the ability to capture very large numbers of high-quality digital images quickly and cheaply. The number of digital images that are being taken by the average consumer each year is growing significantly.

Coinciding with this, there has been a recent explosion in the popularity of 'Web 2.0' style social networking sites. Hundreds of millions of users worldwide, possess accounts in a myriad of different types of online networking websites such as facebook.com and bebo.com. One popular attribute of social networking sites is the storing and sharing of digital imagery. Many users of social networks routinely upload and share photos with their friends and sometimes much larger

social circles. Another popular facet of the 'Web 2.0' revolution is the emergence of online photographic repositories. Web sites that store and organise personal image collections online such as Flickr [Flickr, 2004] have very large volumes of personal images in their databases. Flickr currently has over five billion personal photos stored online with an average of 3-5 million images being uploaded daily. Unfortunately, the proliferation of shared photographs has outpaced the technology for searching and browsing such collections. With this very large body of growing information, there is a clear requirement for efficient techniques to structure and organise it, and for new and novel ways to present this information to users.

Many consumers, tourists in particular, capture large numbers of images in destinations that they visit, and upon return, share these images online with friends and family. One large genre of images that are being uploaded to online image repositories, are photographs containing famous landmarks from around the world. Due to drawbacks in image classification technology, in most cases it is not possible to automatically classify high level semantic information from these images (such as to label them with the name and location of a landmark) based on image content alone.

For high-level semantic image retrieval queries, retrieval systems are forced to rely on text based retrieval methods based on captions created by users, with little or no formal rules on objectivity or detail. This can lead to retrieval errors due to homogeneous and subjective captions, and in some cases no caption provided at all. Homogeneous captions result in poor reliability of individual items in search, and subjective labels are unlikely to be useful for users other than the captioner, or for the captioner themselves searching for the image in a different context. Homogeneous captions are observed to be a common occurrence where a user uploading a large number of images will use the same caption to describe the

whole set, which creates obvious problems trying to distinguish between images in this set based on text alone.

The average consumer, taking a picture with their digital camera or smartphone generally does not pay much attention to how images are stored, organised and retrieved. They simply want a fast and reliable automated technology that allows them to photograph an image and at a later stage retrieve, view and share that image. They don't wish to spend large amounts of time, in what they regard as the monotonous task of providing textual descriptions for images before uploading them to a web site of their choice. Therefore, an automated approach to this task is desirable.

The main motivation behind this work is to create a reliable framework for the automated classification of popular landmarks using technology that will soon become commonplace for the average user of digital photography tools. Geographical features and landmarks have long been one of the most commonly photographed objects that tourists capture and commonly search for in image retrieval systems. Sanderson and Kohler [Sanderson and Kohler, 2005] claim that almost one fifth of all web search engine queries had some geographical relationship, while Gan et al. [Gan et al., 2008] claimed that one in eight web queries contained the actual name of a specific location. While in the past, it was very difficult to reliably retrieve images of landmarks, several advances in technology and changes in the manner in which consumers utilise technology have now provided the means to do so.

Several advances in computer vision have enabled automated, accurate matching of images, even when using very large sample data collections. This has been due to the development of effective methods for detecting image features with a high level of repeatability. Additionally, several discriminative approaches to describing these image features have been proposed, enabling a method to

distinguish between images and matching similar ones with a high degree of accuracy.

Another important technology that has recently been commercialised is the Global Positioning System (GPS). GPS receiver devices can now be found in electronics stores, department stores and even in supermarkets worldwide. Globally, consumers have embraced this technology to the extent that it has changed the ways that people carry out many tasks, such as hiking, driving a car, and in many cases, taking photographs. GPS receivers can now also be found in many smartphone devices that make up a large portion of the total mobile phone market. The main advantage of GPS receivers from the perspective of image classification or retrieval, is the ability that GPS offers to associate a geographical location with an image. In image matching or retrieval tasks, geographical information allows pruning large numbers of non-candidate images from a training dataset. Many image matching algorithms that were previously computationally infeasible can now be achieved in the reduced search space that becomes possible by introducing geographical data.

Smartphone technology has become significantly more advanced in recent years. Many mobile phone manufacturers now produce smartphone devices that come equipped with high-quality digital cameras, along with GPS receivers and fast 3G and soon to be 4G internet connections. The combination of these technologies allow for the capture of imagery, followed by the automated geotagging of this imagery, and the immediate upload of selected images to online photo repositories. Images may be uploaded with or without textual annotation. As noted earlier, even if such annotations exist they may be very sparse.

The framework that is hypothesised in this work aims to provide an automated solution to the problem of providing accurate textual metadata for images containing large landmarks. Such metadata can potentially support image search and automated captioning applications. Due to the rapid proliferation of smart-

phones worldwide, the aim is to create a framework that would make it possible to capture an image on a mobile device, have the device automatically tag the image with an accurate and semantically relevant textual description, and then upload that tagged image to a online website. This reduces the human effort required to manually provide textual descriptions. This automated creation of tags would have to take place in near real-time as users generally are reluctant to use technologies which are considered time-consuming or inconvenient. In this work a framework that aims to meet these criteria is outlined, implemented and evaluated.

1.3 Hypotheses

In this research into automated recognition of landmarks within community contributed datasets, the aim is to test two main hypotheses, both of which are stated below.

- **Hypothesis 1.** *It is hypothesised that by structuring image data into semantically and visually related groups, that it would be possible to create a memory efficient framework based on machine learning algorithms to accurately classify commonly photographed landmarks within geo-tagged image corpora in real-time* A framework system is proposed and implemented to investigate this hypothesis in a large dataset of community contributed images. An extensive investigation is to be carried out using a variety of established computer vision techniques fused with machine learning approaches to test this hypothesis.

- **Hypothesis 2.** *It is hypothesised that by combining a machine learning based method with a commonly used tree indexing based approach that it is possible to improve upon existing methods to classify landmarks within digital images in a memory efficient manner* To test this hypothesis, a hybrid approach is proposed

based on the fusion of machine learning methods with a tree structure for indexing visual features to allow for the classification landmarks

1.4 Objectives

The primary research objectives in this work are to create a methodology to automatically recognise images containing landmarks in a time and memory efficient manner based on community contributed data. Based on landmark recognition using these datasets, it may be possible to provide descriptive and accurate captions for images containing large landmarks. The aim is to achieve this using a novel approach based on established computer vision techniques, such as local image feature matching and methods for quantising local image features into global feature vectors that are suitable to be used as inputs into machine learning methods. This approach combines two techniques for image classification. The first technique is based on clustering images of landmarks taken from similar viewpoints and using machine learning techniques to classify test images based on features extracted from these clusters. This classification approach allows for the efficient reduction of search space when classifying a query image. The second technique is a divisive hierarchical approach that filters out non-candidate images using an efficient matching process, before finding an accurate match using more expensive image matching techniques. It is hypothesised that this automatic classification could be carried out in a timeframe, that allows for interactive addition and searching of a large scale database of geo-tagged images. It is also intended to evaluate how well this approach will perform against a current state of the art approach using a large image corpus, harvested from online community sources.

As part of this framework, one problem that must be solved is how to accurately cluster, such a large scale dataset into visually similar clusters within a

feasible timeframe. Brute force point to point matching of local image feature descriptors generally performs quite well in terms of accuracy, when clustering images on a small scale. With large scale datasets containing perhaps millions of images however, it becomes computationally infeasible. The k-means algorithm is traditionally very slow (complexity of O(KNM) where K is number of iterations, N is the process of re-assigning cluster centres and M is the process of calculating vector distances) and therefore an alternative clustering algorithm must be utilised that reduces the time required to process the k-means algorithm while still retaining accurate clustering results. Another aim of this work is to research several multi-tiered approaches that will first utilise low cost image features that might not discriminate as well as interest point features, but are much faster to compute and compare before carrying out more accurate but costly interest point matching. These image features are extracted and compared in the early stages of the clustering approach. This is followed by interest point descriptor comparisons on a filtered set of images. In this investigation, it will be imperative to find the right balance between speed and clustering accuracy, as it is envisioned that this framework could theoretically be utilised on a very large scale collection of geo-tagged images.

Another objective is to analyse how best to train the Support Vector Machines (SVMs) to create robust classification models. What granularity of clustering will perform most accurately? Would images taken from multiple viewpoints of a landmark with more training data perform better than training images from just one viewpoint of a landmark. Many other combinations of parameters and features must be explored to create the most robust and accurate SVM models for this purpose, eg. What type of SVM kernel is most suitable to use for this purpose? What values should be used to assign to errors in the training phase? Would indiscriminate global based, low-level features improve or harm SVM classification? It can be difficult to find the right balance of features and parameters to use to train

a robust and accurate classification model, and the wrong combination will lead to a noisy and inaccurate classifier. Additionally, another objective is to analyse an alternative machine learning algorithm to conclude what might perform best in this work. The aim is to experiment extensively to ascertain what combination performs best for the problem described in this thesis.

Secondary research objectives include analysing the accuracy of community contributed metadata for the purposes of image matching and classification. It is important to know the limitations of the metadata that is available for the purposes of organising and describing collections of images. It is not useful to successfully match an image to images in an annotated training collection, only to tag that image with an incorrect caption or set of tags that are associated with the matched samples. It is aimed to analyse how useful the tags that accompany community data might be, and how best to extract correct tags from successfully matched images.

As the majority of images within the corpus have been manually geo-tagged by the uploader, it is possible that large amounts of the corpus will contain inaccurate location information. The accuracy of the geographical data is also an important consideration. The geographical pruning and clustering processes are based around disregarding images located outside a geographical radius. It is necessary to ascertain what is an appropriate spatial radius to adopt during the clustering and classification procedures to ensure maximum speed and accurate classification. Will too small a spatial radius eliminate potential candidate images? Will too large a spatial radius render the proposed approach infeasible due to inefficient pruning of non-candidate images? The objective is to analyse the geographical information with the hope of ascertaining the best trade-off between classification speed and classification accuracy.

The final part of the landmark recognition framework is to automatically provide a textual annotation to a query image based on the retrieved matched

images from the corpus. In a professionally annotated corpus, this process is trivial as it means simply selecting the metadata associated with each matched image. In a community contributed dataset, however, it becomes very difficult to achieve this task with a high degree of accuracy. Given that large numbers of community provided text tags will be heterogeneous or semantically irrelevant, it remains a challenge to automatically create semantically relevant subsets from sets of community provided tags. A secondary research objective of this thesis is to implement and evaluate techniques that will help to solve this problem and improve upon the current state of the art methodologies.

1.5 Structure of Thesis

The remainder of this thesis is structured as follows.

In chapter 2 the background to the research topic is introduced. An introduction to image retrieval is first provided, describing historical background to the field including some of the early technological innovations and established practices in image retrieval systems and techniques developed and evaluated over the years. More advanced approaches to classifying semantic information from image content in the context of image retrieval are then discussed. Next a number of techniques for comparing and classifying images using local image features are introduced and described. The chapter concludes with an overview of previous work carried out in the research field of object and landmark classification in image collections, along with similar fields. It is important to provide an overview of the background to the work presented in this thesis, as without these previous technological breakthroughs, it would not now be possible to provide a framework for large-scale accurate image classification.

In chapter 3 the concept of social networks and community contributed data collections are introduced. An overview is provided, describing data that is currently available to augment the implementation and analysis of landmark classification methods. The advantages and disadvantages of community contributed metadata are outlined and an analysis is carried out to determine how accurate and useful this data might be for the purposes of the work described here. The approach that is used to create the dataset, which in turn is used to build the classification framework, is discussed, followed by a description of all datasets utilised. An analysis of the accuracy of a subset of Flickr geo-tags is also carried out, along with an analysis of the relevance of the manually created textual descriptions of image content. The chapter concludes with an evaluation of community contributed data.

In chapter 4 the concept of clustering visually similar images is introduced, along with the importance of an efficient approach when clustering large scale image corpora. This is followed by a brief description of clustering algorithms, in particular hierarchical clustering. The first clustering approach evaluated in this work, based on community provided text is described, and the effectiveness of this approach is examined. A number of different approaches to hierarchical clustering using low-level image content features are then reviewed. An alternative clustering approach that is based on hashing techniques for quickly finding approximate nearest neighbours in feature space is also implemented and evaluated. The chapter concludes with a section on hierarchical combinations of low-level features, interest point features and contextual features. The results of these combinations are analysed and described in detail.

In chapter 5 the core hypothesis of the thesis is introduced and the core motivation behind this work is described in detail. The chapter starts with a description of machine learning and many of the commonly used approaches to classifica-

tion, in particular the machine learning method, Support Vector Machines. The problem of using local image features for classification purposes is described, and alternative approaches based on the quantisation of multiple local image features into global based vectors are introduced. An evaluation is carried out to ascertain how successfully machine learning classification models can be trained to recognise landmarks using manually created training sets. The chapter concludes with the results of an in-depth investigation and evaluation of utilising machine learning methods for the purposes of landmark recognition.

Chapter 6 investigates approaches to landmark classification in situations where classification models are not applicable. This occurs mainly in situations where an individual landmark is sparsely represented within a corpus and there is insufficient data to train a robust model. Two main approaches are investigated, a vocabulary tree based approach and an efficient hierarchical classification scheme. A hierarchical based approach that makes use of a scene classification methodology, is introduced, and many different forms of this approach are evaluated. This hierarchical approach utilises several low-level image classifiers, and an analysis of these classifiers is presented. Following this, the background of a vocabulary tree structure is introduced, along with motivations behind its use. A novel hybrid method combining SVM models and a hierarchical classification approach is introduced and presented. This approach combines the use of classification models with a hierarchical classification pipeline that aims to classify popular and non-popular viewpoints of landmarks. An implementation of the current state of the art method for landmark classification is introduced and evaluated against the methods that make up the framework presented here.

In Chapter 7, an introduction is provided outlining the issues that arise with captioning images using community contributed metadata. Previous work in the field is introduced along with a explanation of proposed approaches to improve

on this work. An analysis of these approaches is carried out and the chapter concludes with a results of this analysis.

Finally in **Chapter 8**, the conclusions of this work are presented referring back to the original hypotheses and research objectives. This chapter concludes by outlining various research avenues that could be explored in future work.

Chapter 2

Content-Based Image Retrieval and Landmark Classification

This chapter provides a background to image information retrieval and image classification. It gives a review describing previous research in the field, followed by a discussion on current state of the art approaches to the automated recognition of large landmarks.

Image retrieval is a very large research field, with many significant research groups working worldwide on different areas within the field. Advances in image retrieval have drawn different techniques and expertise from many other research fields including text retrieval, computer vision, psychology and geographical information science among many others.

In this chapter, the aim is to review the history of image retrieval, the many technological breakthroughs that have been made in the field, and the difficulties that researchers are faced with. The objective is to provide the reader with contextual information concerning the research problem addressed in this thesis. This should also help the reader gain insights into challenges posed by accurate, automated captioning of digital imagery.

2.1 Introduction

The task of automatically presenting images to a user that are relevant to their wants and needs is called *image retrieval*. Image retrieval is currently a very active research field, mainly because of the large amounts of digital imagery that are now being created and stored by consumers. Most image retrieval systems accept a user's query and, using an algorithmic method attempt to return images that are most relevant to the query from an available corpus of images. The meaning of relevance can be subjective. For example, a user might associate temporal similarity as a measure of relevance between two images, such as two images photographed during the course of a holiday and therefore could be considered relevant, even though they share no visual characteristics. Although there have been methods suggested to retrieve images based on other measures of relevance (such as event detection), for the entirety of this work it is assumed that relevance is correlated with visual similarity.

Image retrieval techniques can be roughly categorised into two main approaches:

- **Content-Based Image Retrieval.** Content-based image retrieval is the organisation of sets of images and the retrieval of relevant images from these sets based upon the actual visual content of an image. Content-based retrieval may refer to different measurements of colour, texture or region shapes among others that are created from the image at a pixel level. More recent approaches focus on the comparison between smaller regions within an image, rather than considering the entire image as one entity.

- **Context Based Image Retrieval.** Context based image retrieval is based upon information that is available about an image, commonly referred to as contextual metadata. This information can include manually created text tags describing the content of the image or information based in the

Exif header of the image. Most large scale image retrieval systems in use today are based on context-based image retrieval with each image being represented by a small number of keywords that have been provided by a human annotator.

2.2 Content-Based Image Retrieval

Although the techniques applied in this framework rely on contextual information, the majority of the research is carried out in the field of content-based image retrieval (CBIR). CBIR is a large research field concerned with the retrieval of images based on their pixel content. It combines technologies and methodologies from many different research domains, such as computer vision, machine learning, information retrieval, data mining, statistics and psychology. Several advances in the field have enabled the creation of commonly used applications of CBIR and have led to a wide interest in the field in recent years. It has been shown that CBIR and related fields have grown roughly exponentially in terms of the people involved and publications since the year 2000 [Datta et al., 2006]. CBIR is a relatively mature research field with many solutions to problems have been suggested and researched, however it is by no means considered a solved problem. Large scale semantic retrieval as a real world technology is still a long way off. In 2000, Arnold Smuelders outlined in a journal article called 'Content-Based Retrieval: The End of the Early Years', the main research problems facing CBIR as the 'Sensory Gap' and the 'Semantic Gap' [Smeulders et al., 2000].

- **Sensory Gap.** Smeulders describes the sensory gap as 'the gap between the object in the world and the information in a numerical/verbal/categorical description derived from an image recording of that scene'. He outlines several problems that exist when trying to process the visual information available in the real world into a relevant computational description. For

example, when a digital photographic device takes an image, it is stored as a numerical representation. This representation is significantly quantised from the amount of information that is available in a scene. The level of this quantisation is based on limitations with individual image capturing devices and can create problems with image recognition. Other sensorial problems that occur with image capturing include image noise and object occlusion.

- **Semantic Gap.** In terms of image retrieval from a user's perspective, the 'Semantic Gap' is a more important research issue. The 'Semantic Gap' refers to the inability of computers to classify high level semantics that a user might interpret from the content of an image. Computers excel with numerical queries, such as retrieve an image that contains 20% red pixels, 30% blue pixels and 50% yellow pixels, but struggle to accurately return images based on semantic human queries, such as return an image of 'a red motorbike', 'Guns and Roses' or return an image of 'an argument'.

2.2.1 Challenges in Content-Based Image Retrieval

Traditionally in image retrieval, images are represented by feature vectors of numerical data, intended to correlate with different features and attributes of the image. These feature vectors are then used in conjunction with different retrieval models which seek to return images relevant to user's information needs. Due to the large variations that can occur between apparently similar images, several challenges exist when using computer vision techniques for classification purposes. Some of the variations that can greatly effect classification performance are:

- **Illumination.** Small changes in illumination might not cause problems for humans to distinguish between similar scenes, but an image feature

extraction algorithm might represent similar scenes very differently due to small variations in illumination between the images. This is particularly problematic when using image features that are calculated based on colour information, or changes in image intensity values (changes in brightness). Changes in illumination may be caused by the time of day that an image was taken, different weather conditions, and different camera settings. These possible illumination changes must be taken into account when developing algorithms to extract image features or creating classification models to recognise different image categories.

- **Orientation.** The orientation of the image can cause problems when retrieving and classifying images. Many feature extraction algorithms will detect features differently, based on the orientation of the image at extraction time. Several image feature algorithms will also describe an image feature based on the orientation of detected features. Clearly this presents a problem when trying to match visually similar imagery, as similar images with small orientation differences might be described differently and therefore will not be successfully matched. Techniques have been developed to solve this problem, and several algorithms based on these techniques have been shown to be invariant to large changes in orientation [Bay et al., 2006][Lowe, 2004].

- **Scale.** Scale refers to the distance between an object in an image and the camera. From a semantic perspective, two pictures of a building facade taken from similar viewpoints but from different distances are the same and should be classified as such. It is imperative for an information retrieval system to be able to recognise similar objects and locations taken at various scale levels.

- **Occlusion.** Occlusion is when an object or objects are blocking or partially blocking the intended object to be photographed. Occlusion occurs when

another object is situated in the viewplane between the location of the camera and the main object within an image. Several common types of occluding objects include foliage and people and transport vehicles passing in front of a camera while an image is being taken. The main problem with occlusion is that image features are describing the occluding object as well as the main scene or object. Depending on how severe the occlusion, the accuracy of any retrieval method will be reduced.

- **Affine Variations.** Affine variations arise from differences in image viewpoints. Images of an object with different lines of sight will look different. Small affine variations do not effect a human observer who can still recognise the same object from different, albeit similar viewpoints. When using computer vision algorithms, the same variations can produce completely different image features.

- **Intra-Class Variation.** Intra-class variation refers to the differences in visual similarity that can occur between different instances of the same class of object. For example, two buildings might have different sizes, colours, shapes and features, however it is still possible to recognise that an image contains a building due to similar characteristics that appear in the majority of buildings.

- **Inter-Class Variation.** Inter-class variation refers to the visual similarities that can exist between different classes of objects. It is possible that two landmarks located within a single city could be visually very similar. This particularly prevalent when different geographical objects within a city are built in the same architectural style.

2.2.2 Low-level Image Retrieval

Low-level image retrieval is based upon global measurements of different types of image attributes gathered from analysing the raw pixel data within an image. Low-level image features can be split into three main categories;

1. Colour Based Features
2. Texture Based Features
3. Shape Based Features

Many of the first successful image retrieval systems were based solely on low-level image features. The Query by Image Content (QBIC) image retrieval system [Ashley et al., 1995], developed by IBM was one of the earliest image retrieval systems. The QBIC system was based on combinations of colour, texture and shape features to return images to a user, based on a "query by example" interface, which is an interface where the user enters some input such as a test image, or actually draws the desired query image into a canvas object and the system returns images visually similar to the examples. The system has been successfully integrated into a number of different domains, such as an art retrieval system for museums and galleries, along with stock photo retrieval and applications in the textile industry [Petkovic et al., 1996]. The idea behind the QBIC system was to retrieve images based on measurable properties such as colour and shape. For example, the QBIC system allows for users to draw the shape of a car into a user interface screen. The system would then create a numerical representation of this car shape and search for other similar representations in their database. This task is very different from the more advanced semantic query 'Retrieve all images of cars', which is a high-level semantic query that requires a high-level understanding of the content of an image, which does not necessarily correlate with visual similarity.

Another early image retrieval system was the VisualSEEK system developed at Columbia University [Smith and fu Chang, 1996]. The VisualSeek system retrieves images based mainly on colour features, specifically features generated from visually salient regions of an image. The system allows for the retrieval of images based on region based queries, whereby images are retrieved based on the spatial layout of colour within corpus imagery.

Low-level image features can typically provide an efficient means to retrieve visually similar imagery from a large corpus, as most low-level features are global based and tend to be inexpensive to extract and compare.

2.2.3 Types of Low-Level Image Features

Colour

One class of low-level image features are colour features. These are based on different representations of the colour values within an image.

The most basic image colour feature is the histogram, which is still widely used today. A histogram is a description of the frequencies of pixel intensities and colour values in an image. Each bin in a histogram represents the number of pixels in an image that will have a value corresponding to the bin value eg. histogram H for a grayscale image I that has an intensity value range of $I(u, v)\epsilon[0, K-1]$ with K=256, will have 256 bins. H(i) represents the number of pixels in image I with the intensity value i. A colour histogram in thee RGB colour space represents more information than a greyscale histogram. An average RGB colour histogram will have 768 bins with 256 bins per colour (Red, Green and Blue).

RGB histograms provide an effective measurement of colour distribution, however, a significant issue for image matching using colour features is that they are not robust in particular to changes in illumination [van de Sande et al., 2010]. Even small changes in illumination can cause large changes in the colour distribu-

tion, which can impair matching of apparently similar images. It has been shown that histograms processed in different colour spaces outperform RGB histograms for many computer vision and image retrieval tasks [Borghesani et al., 2009]. For example, histograms based on the Hue, Saturation and Value (HSV) colour space are more similar to the way a human perceives the visual spectrum and hence outperform RGB histograms for visual retrieval from a human perspective [Kotoulas and Andreadis, 2003] [Borghesani et al., 2009]. Other types of histograms have also been tested for image retrieval purposes in a number of different colour spaces, such as the LCH [Missaoui et al., 2004] and the YCbCr colour spaces [Talbar and Varma, 2010]. Traditional histograms tend to be susceptible to noise interference within an image, which is a phenomenon consisting of random light variations captured by a camera sensor. Stricker and Orengo [Stricker and Orengo., 1995] proposed an alternative to the traditional histogram, by utilising cumulative frequencies of intensity values to represent a histogram. They proposed the cumulative colour histogram, in order to add robustness to image noise, and demonstrated that this approach outperformed the standard colour histogram method for retrieval tasks based on colour features.

As the CBIR research field grew, more and more groups started to exploit colour information in different ways. The main drawback of traditional colour histograms is that they do not capture any spatial information about the distribution of colour within an image. Several algorithms have since been developed that measure distributions of colour and additional measurements containing information about the spatial layout of colour values within an image.

Birchfield [Birchfield and Rangarajan, 2005] introduced a spatial colour feature called a spatiogram, that has been used successfully in object localisation tasks [Conaire et al., 2007]. The spatiogram is an extension of the standard colour histogram structure in that it adds statistical information about not only the distribution of colour within an image but also can add information about the spatial

relationships between these distributions. As part of the MPEG7 standard various descriptions of colour features were proposed, and have been used in a wide variety of retrieval and classification tasks, from retrieving video surveillance imagery [Annesley et al., 2005] to adult image classification [Kim et al., 2005]. Many of these MPEG7 features contain spatial measurements such as the colour structure feature [Messing et al., 2001] and the colour layout features [Kasutani and Yamada, 2001]. Using these improved methods to provide complex descriptions of colour distribution within an image allows for more efficient comparison of images, using in most cases features that have been quantised into small feature vectors, augmenting the fast comparison of large numbers of images.

In this work, different colour features are utilised as higher level methods in a hierarchical clustering scheme in Chapter 4.

Texture

It is quite difficult to provide a strict definition for image texture. Image texture could be described as a measure of reoccurring patterns within an image, or perhaps a measure of the small variations in changes of intensity within an image region or as a global measure. In general, texture provides extra information about the spatial layout of intensity levels within an image.

First order statistics such as mean or standard deviation of pixel or intensity values are generally not useful measurements to calculate image texture as many visually different images might have similar or identical values. Therefore, the first types of image texture descriptions consisted of a set of measurements based on second-order statistics. In 1973 Haralick proposed an approach to extract texture properties from image blocks using grey level co-occurrence matrices (GLCM), which are a measure of the frequencies of the spatial relationships that exist between certain pixel values [Haralick et al., 1973]. Five texture measures were proposed:

- Energy = $\sum_i \sum_j N_d^2[i,j]$
- Entropy = $\sum_i \sum_j N_d[i,j] \log_2 N_d[i,j]$
- Contrast = $\sum_i \sum_j (i-j)^2 N_d[i,j]$
- Homogeneity = $\sum_i \sum_j \frac{N_d[i,j]}{1+|i-j|}$
- Correlation = $\frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_d[i,j]}{\sigma_{i\sigma_j}}$

where N is equal to the GLCM for an image. At the same time, Tamura et al. [Tamura et al., 1973] proposed a similar approach to describing texture, based on human perception. This approach was composed of 6 features; Coarseness, Contrast, Directionality, Linelikeness, Regularity and Roughness. While Tamura's and Haralick's features were widely used for early image retrieval tasks including landmark classification [Takeuchi and Hebert, 1998], they are not discriminative enough for large scale image retrieval tasks, and have since been outperformed by more advanced texture features [Howarth and Rger, 2004]. While they were used initially for image retrieval purposes, they lack the discrimination values to be effective for large scale image retrieval tasks, particularly when not used in conjunction with other sets of features.

Due to the limitations of these second-order statistical features some more advanced texture features have been proposed, such as those calculated from distributions of image edges. Edges are points within an image where there are measurable sharp changes in intensity in one or more directions. Several algorithms have been developed in the computer vision and machine visions fields to detect the presence of edges, such as the Canny algorithm [Canny, 1983]. As edges correspond to sharp changes in intensity values, they can be used as a rough measure of image texture and several image features have been developed

for this purpose such as edge orientation histograms [Gagaudakis et al., 2000] [Jain and Vailaya, 1996] and the MPEG7 edge histogram [Manjunath et al., 2001].

Another alternative to edge based texture features, are features based upon wavelet transform functions, such as Gabor filters. Gabor texture features are built around banks of Gabor filters and have been used in many image analysis and image retrieval tasks, such as iris recognition for biometric identification [Daugman, 1993] and object segmentation [Jain et al., 1997]. Each Gabor filter captures change at a specific frequency in a set direction. Generally a texture feature will contain dozens of these filters containing a number of varying frequencies and directions, to give an overall representation of image texture. These features are described in more detail in Chapter 4.

Over the years, more and more advanced methods to describe texture within an image have been proposed by the research community, and image features based on texture continue to play an important part in many image retrieval [Manjunath and Ma, 1996], image clustering [Cai et al., 2004], image classification [Peterson and Larin, 2009], image segmentation [Weldon and Higgins, 1999] and image matching tasks [Zhang and Kosecka, 2007].

In this work, texture features are utilised and their performance is evaluated as part of image clustering processes in Chapter 4, along with pruning non-candidate images in a hierarchical classification technique in Chapter 6.

2.3 Low-Level Semantic Classification

As the extraction and representation of image features became more reliable, several techniques were developed to classify low-level semantic information from an image. Combinations of global low-level image features can be combined with classification techniques to infer basic information about the content of an image. For example, in the absence of EXIF information, colour based image

features can be useful to determine whether an image was taken during the day or at night [Kuthan and Hanbury, 2006]. Successful low-level classification of semantics allowed for image retrieval systems to organise and return images based on more humanistic queries. For example, instead of returning images consisting of 30% red pixels and 70% blue pixels, a system could now be queried to retrieve images containing a sunset, snow covered landscape or perhaps a seascape scene. These types of semantics are getting closer to the types of queries that humans might make to a retrieval system.

Several successful classification methods were developed to recognise a variety of low-level semantics such as the recognition of a cityscape (urban) or landscape (rural) scene [Yan et al., 2003]. Szummer and Picard [Szummer and Picard, 1998] combined colour histograms with texture features to train a nearest neighbour classifier to recognise whether an image was taken indoors or outdoors. Vailaya et al. [Vailaya et al., 1998] trained a k-Nearest Neighbour classifier to group images into a finite number of low-level semantic classes. These classes contained cityscape, landscape, forest, mountain and sunset among others. They used colour and edge features as inputs into the classifier.

As more research groups began to experiment with image semantic classification techniques, it became possible to infer higher level semantic knowledge from an image. Several mid-level semantics were successfully classified from images. One established mid-level semantic classifier that has been heavily researched and is in widespread use today is the detection of human faces within an image. Early approaches to face detection involved searching for regions of an image with large areas of colour similar to that of human skin, and analysing these regions to determine if they contain a face [Hsu et al., 2002], [Singh et al., 2003]. Once these regions are detected they are compared against a database of face images to measure the correlation. More advanced techniques were also developed that allow for the real-time detection of faces within images. Viola and Jones

[Viola and Jones, 2001] developed a face detection technique that made use of an image representation called *integral images* combined with fast approximations of Gabor filters, which allowed for the fast accurate classification of faces at 15 frames per second. They utilise a cascade of classifiers, where simple classifiers are used to reject candidate regions for faces, while more complex classifiers are used at later stages to achieve low false positive rates.

For the investigation described in this thesis a suite of low-level classifiers was trained to aid in a hierarchical approach to image classification. A full description of the implementation and performance of these classifiers can be found in Chapter 6.

2.4 Context-Based Image Retrieval

It has been shown that image content alone is not sufficient for high level semantic classification of imagery . The approach adopted today in most large scale image retrieval systems is based upon textual information associated with an image. Short textual descriptions called tags are created by users while uploading images into a system. These tags are intended to represent the semantic content and context of the image. The system indexes these short textual descriptions and retrieval is carried out using established text retrieval techniques.

Several other approaches to context based retrieval have been implemented and tested. O Hare et al. [O'Hare et al., 2005] developed a system called MediAssist with the aim of annotating people in personal photo collections in a semi-automated manner. The MediAssist system used different types of context data such as the time an image was taken and camera settings available in the EXIF header. The system classified images into semantic categories based on this context information. Images were grouped into four different categories of light status; Daylight, Dusk, Darkness and Dawn, using the temporal information

available. This was used to gather weather data, which combined with the GPS allowed the system to determine ambient light conditions. They also developed a technique to classify an image as indoor or outdoor based on a number of camera settings such as brightness, shutter speed and the ISO values. All images within the MediAssist system contained GPS information and images were also grouped and classified based on location.

2.5 Local Image Features

Local image features focus on salient regions within an image. These are regions that display a certain amount of non-uniformity in intensity values. These features can be used to find correspondences (visually similar image patches) between sets of images and tend to be more discriminative than low-level features, such as those based on edges, colour and texture. Since local image features are based around small regions centred on informative image regions, they also tend to be far more robust to background clutter and occlusions. These features, also called interest points, have been used in many applications of computer vision including object recognition [Lowe, 1999], object recognition in video [Sivic and Zisserman, 2003], object classification and image retrieval from large databases [Philbin et al., 2007].

Ideally, interest point algorithms should be invariant to scale, translation and rotation. The algorithms should also be partially invariant to small affine changes and changes in illumination. The most important feature of an interest point detection algorithm is repeatability, which is the ability of the algorithm to detect the same interest points in different images.

The detection and description of salient regions is now relatively mature and several algorithms exist that can detect salient regions and create a highly discriminative feature descriptor to describe the region. One of the first, and still widely used, interest point detectors is the Harris corner detector [Harris and Stephens, 1988].

Figure 2.1: An illustration displaying an example of the discriminatory attributes of local image features. In this example, three visually and semantically similar images are selected from the test collection, used in this work. Three of the images contain visually similar objects in a similar setting, however only two of the images contain the same object. The test image is matched against a relevant and non relevant image using the SIFT algorithm [Lowe, 2004]. There were 72 correspondences found between the test image and the relevant image, while there were zero correspondences found in the visually similar non-relevant image.

The Harris algorithm detected points in an image that were located on corners (horizontal and vertical edges). Several commonly used interest point algorithms include Multi-Scale Oriented Patches (MOPS) [Brown et al., 2005], Scale Invariant Feature Transform (SIFT) [Lowe, 1999] and Speeded Up Robust Features (SURF) [Bay et al., 2006].

2.5.1 Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT) algorithm was first published in 1999 [Lowe, 1999]. It has since become one of the most widely used algorithms to detect and describe local image features. The SIFT algorithm provides a method for detecting distinctive, salient regions within an image (interest points) and for creating highly discriminative feature vectors for each of these interest points. These feature vectors can then be used for reliable matching of visually similar interest points within different images. An example of this is depicted in Figure 2.1.

The first stage of the algorithm is to select locations/regions within the image that could be suitable as candidates for interest points. Repeatability is the important factor in this stage. It is important that the same locations would be selected in a similar image. A scale-space pyramid [Lindeberg, 1994] is created for each image. Difference of Gaussian functions are then applied to each image in the pyramid and the local minima and maxima are selected as candidates for interest points as illustrated in Figure 2.2.

The location and scale of each candidate point is then determined. Unstable interest points are filtered out based on a number of measures such as low-contrast and keypoints that are located on strong edges within the image. An orientation is assigned to each keypoint based on the image gradients surrounding the keypoint location. The SIFT descriptor is calculated based on local image gradients around a keypoint. Orientation histograms are created over 4×4 sample regions around the interest point. This creates a 4×4 array of orientation histograms each containing 8 different orientation directions thus creating an image descriptor of length 128 ($4 \times 4 \times 8$).

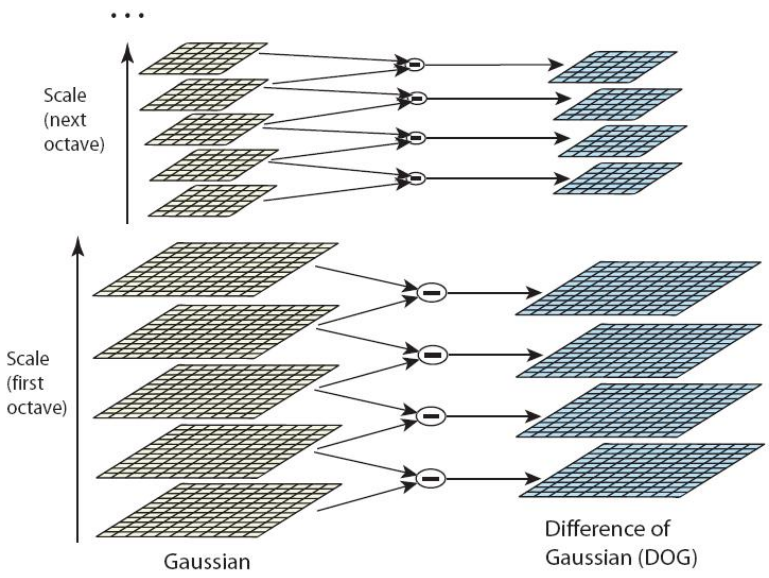

Figure 2.2: An illustration of a scale space pyramid using difference of Gaussian functions that make up the scale space extrema detection in the first part of the SIFT algorithm taken from [Lowe, 2004]

2.5.2 Speeded Up Robust Features

Speeded Up Robust Features (SURF) is an algorithm created by Herbert Bay et al. [Bay et al., 2006] to detect and describe salient regions within an image. The SURF algorithm is similar to Lowe's SIFT algorithm [Lowe, 2004] in that the algorithm will detect 'regions of interest' or 'interest points' within an image, however it is optimised to detect these regions in a shorter time frame than SIFT.

One of the big advantages of SURF in comparison to SIFT is speed. SURF image features are detected very quickly due to the algorithm's use of integral images. SURF image features are also faster to match and compare, as they are half the size (vector size of 64) of SIFT features (vector size of 128). This reduction in the size of the descriptor does not however, harm the discrimination properties of the features. Like SIFT, SURF is invariant to scale, rotation and small variations in image viewpoint (affine variations) [Juan and Gwon, 2009]. Several studies show that the SURF algorithm performs at least as well, or better

than the SIFT algorithm for certain image matching tasks [Connaire et al., 2009], [Murillo et al., 2007]. The SIFT algorithm, however is more invariant to changes in illumination.

Another big advantage with the SURF algorithm is the inclusion of the sign of the Laplacian within the feature vector. This distinguishes bright regions on a dark background from dark regions on a light background, enabling only the matching of bright interest points against similar bright interest points and similar for dark interest points. This effectively halves the number of interest point comparisons required when matching images.

Interest Point Detection

The detection of interest points using SURF is faster to process than SIFT mainly due to the use of an image representation called integral images, introduced by Viola et al. [Viola and Jones, 2001]. Each point within an integral image is the sum of the values between the point and the image origin, which can be represented by:

$$I_{\sum}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{i \leq y} I(x, y)$$

where x and y represent the coordinates of a pixel within image I.

With these integral image representations the process of calculating the area of a rectangular region consists of four operations and processing time is not effected by changes in the size of an image. For example, A rectangular region bounded by the vertices v_1, v_2, v_3 and v_4, the sum of the intensity values is calculated by $v_1 + v_4 - (v_3 + v_2)$.

Using these integral images, the SURF algorithm detects candidate interest points based on the determinant of the Hessian matrix. The idea behind the algorithm is that feature points will be detected at locations within the image

where the determinant of the Hessian is at its maximum. The Hessian matrix is defined at a point p in an image (where $p = [x, y]$) as:

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix}$$

where $L_{xx}(p, \sigma)$ is a reference to the convolution of the second order Gaussian derivative $\frac{\partial^2 g(\sigma)}{\partial x^2}$, also known as the Laplacian of Gaussian (LoG), at point p with a scale value of σ and likewise for $L_{xy}(p, \sigma)$ and $L_{yy}(p, \sigma)$. In [Lowe, 2004] it was found that a significant performance increase in terms of speed can be obtained by approximating the values of these LoGs using Difference of Gaussian functions. The SURF algorithm also approximates the LoG to speed up the processing time, albeit through the use of a number of box filters.

The combination of box filters with integral images allows for a significant speed increase. For example, with a filter of size 9×9 (which is the approximation of a Gaussian with σ = 1.2, the smallest scale examined with the algorithm), it would require 81 operations to complete a convolution without the use of box filters, whereas, the box filters combined with integral images would require 8 operations. As the σ value increases, the number of operations required without the use of box filters would increase quadratically, while the box filter approach would still require only 8 operations.

Interest points are determined at locations where the local maxima is outputted from the determinant of the Hessian over both area and scale. Bay proposes that the an accurate approximation of the determinant of the Hessian can be calculated using the following formula:

$$det(H_{approx}(p, \sigma)) = D_{xx}D_{yy} - (wD_{xy})^2$$

where D_{xx}, D_{yy} and D_{xy} are the results of approximations of the Gaussians using box filters in the x, y and xy directions, as illustrated in Figure 2.3, and w is a

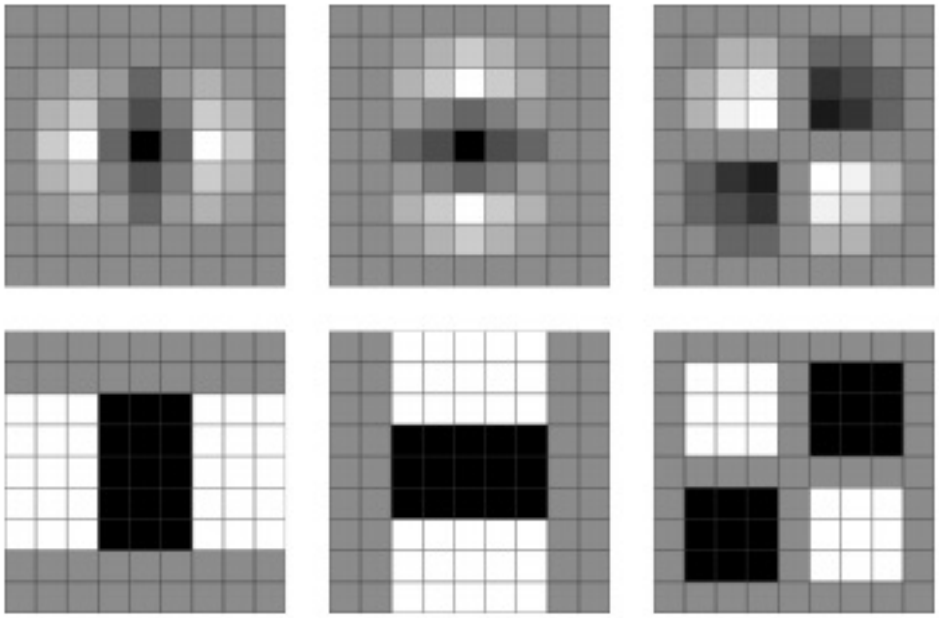

Figure 2.3: An image displaying the box filters (bottom) used to approximate Gaussian functions (top) in the SURF algorithm. These filters are shown in the x (left), y(centre) and xy (right) directions. Image taken from [Bay et al., 2006]

correction constant that is calculated based on the scale and size of the box filters. Bay suggested assigning a constant value of .9 to w.

These candidate interest point regions are then thresholded such that all regions below the threshold value are removed as candidates. The threshold value can be adapted to suit the application with lower thresholds provided more interest points per image but generally the regions will not be as strong (lower level of uniqueness) as with a higher threshold. Non-maximal suppression is then carried out to further filter the candidate regions. Each candidate point is compared in scale-space to its 26 neighbours, which comprise of the 8 neighbouring pixels in the same scale as the candidate and the 9 neighbouring pixels in the scale above and below it. This is illustrated in Figure 2.4. Only points that are a local maximum in this scale space region are retained as candidates.

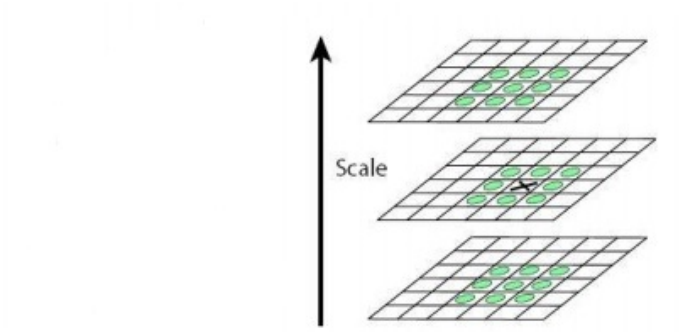

Figure 2.4: **Non Maximal Supression.** A pixel x is marked as a maxima if it is greater than all of it's neighbours in it's scale space interval and the intervals above and below it. Image taken from [Lowe, 2004]

Interest Point Descriptor

The first step in the creation of a descriptor is to assign an orientation to the interest point to ensure that it is invariant to changes in image rotation. A circular region is selected around the interest point, from which the orientation is calculated based on Haar wavelet responses in both the x and y directions. These responses can be calculated quickly through the use of box filters similar to those used in the detection phase of the algorithm.

After an orientation is assigned, the descriptors are then built by constructing a square window around the interest point. This window will have a size of 20σ where σ refers to the scale of the interest point. This window is then divided into 4×4 subregions to retain some spatial information. For each of these subregions, Haar wavelet responses are measured at regularly spaced intervals. These responses in horizontal and vertical directions (d_x and d_y) are summed over each subregion. The absolute values ($|d_x|$ and $|d_y|$) are also summed to capture polarity information about the interest point. For each region a vector with a length of 4 is created based on these wavelet responses, yielding:

$$V_{subregion} = [\sum dx, \sum dy, \sum |dx|, \sum |dy|]$$

As each subregion contributes four values to the feature vector, this yields an overall vector length of 64 ($4 \times 4 \times 4 = 64$). The overall image descriptor that represents each interest point feature vector contains 70 values, which consist of 6 localisation values combined with the 64 feature values. The structure of the descriptor is as follows: $[x, y, \sigma, b, \sigma, l, V]$ where x and y are the coordinates of the points location within an image, σ refers to the scale of the keypoint, b represents the corner strength of the keypoint, l refers to the sign of the Laplacian and V is the feature vector that is used to match keypoints [Bay et al., 2006].

Figure 2.5: An image displaying the interest points that have been detected using the SURF algorithm on a picture of the Arc De Triomphe. As can be seen from the image, the majority of the interest points detected are in the salient regions of the landmark, while few interest point features are detected in the uniform regions.

2.6 Spatial Based Context Retrieval

Another research development that greatly assisted in the accurate matching of images from large scale datasets is the use of spatial information. Many different sources of imagery have become available in recent years, and increasingly they come accompanied by geographical information relating to the location where the image was photographed. Spatial based information can greatly aid image classification. For example, the knowledge that an image was photographed in a small geographical area allows for the classification of imagery using datasets of an unprecedented size, as large numbers of non-candidate images can be quickly filtered out in a retrieval task. Several research groups have developed frameworks and prototype image retrieval systems that make use of spatial data associated with images and fuse it with image content data. The Mediassist system [O'Hare et al., 2005] developed to retrieve images based on low-level image features such as colour and texture fused with location data in the retrieval process. Tests carried out with this system showed that location data alone does not retrieve similar images as effectively as image content data by itself. They showed that for image retrieval the best results were attained by combining image content with image location data.

Another image retrieval system that combines image content and context data, is the Photocopain system developed by Tuffield et al. [Tuffield et al., 2006]. The Photocopain system fuses image content data with different types of context data to aid with image retrieval. It uses GPS data along with weather information and other data such as calendar entries and news stories. The system has pre-trained low-level semantic classifiers which also provide automated tags for images such as natural object, artificial object, indoor, outdoor, landscape, cityscape, portrait and group-photo.

Both of these prototype image management systems have shown that the combination of location data associated with image features can greatly improve the precision and recall of image matching and retrieval tasks.

2.7 Object Classification and Landmark Recognition

2.7.1 Object Classification

As image features and matching techniques became more advanced, a lot of work has been carried out in the field of object classification. Object classification is the automated recognition of different classes of objects and different instances of object classes within an image.

Many approaches to object classification fused low-level image features, such as colour and texture with machine learning algorithms. Tsapatsoulis et al. [Tsapatsoulis and Theodosiou, 2009] utilised the MPEG7 feature set to classify a selection of 8 object categories from a dataset consisting of just under 2000 images. From this work, they concluded that SVMs was the most robust method to use for their task.

Fei-Fei et al. [Fei-Fei et al., 2004] adopt an approach based on image intensity patches quantized into small vectors using a method called 'principal component analysis'. These patches are extracted from salient regions of the image using the Kadir and Brady feature extractor [Kadir and Brady, 2001]. They then adopt a Bayesian based approach to classify objects into 1 of 101 object categories.

As local image features became more widely used, more advanced techniques were proposed based on the use of features consisting of quantised sets of local images features. Liu et al. [Liu et al., 2009] use a modified version of the visual bag of words model (described in Chapter 5) to classify a selection of 10 object classes from the PASCAL dataset. This approach adopts a form of query expansion to

emphasise correlated image patches between sets of objects. They report a small performance increase from the use of this technique over the standard visual bag of words model.

Bosch et al. [Bosch et al., 2008] use a spatial based approach to object classification based on pyramids of image features. They use an image feature called 'Histograms of Orientation Gradients' (HOGs), which was first proposed by Dalal and Triggs [Dalal and Triggs, 2005] to detect the presence of humans within an image, to classify an image into 1 of 101 object categories using a SVM. To incorporate spatial information, they use a pyramid structure that extracts HOGs from finer and finer spatial areas of the image at each descending level of the pyramid. Features based on pyramid structures are discussed in more detail in Chapter 5.

2.7.2 Landmark Classification

One facet of object classification is the classification and retrieval of images containing landmarks, differences between multiple views of the same landmark and distinguishing between images containing different landmarks. In the past, it was not possible to create an automated approach to landmark classification, mainly due to the large visual disparity that exists between different types of landmarks and technology constraints.

In this work, the focus is on the creation of a framework that should allow for automated, accurate and efficient matching and tagging of landmarks contained within digital images. Landmarks are considered to be unique man-made objects, or unique geographical features depicted in an image. The focus is on landmarks due to the significant contribution that they make to a large scale public photo repository such as Flickr (eg. Flickr search for 'Eiffel Tower' returns over 450,000 images, Flickr search for 'Empire State' returns over 370,000 images (June 2011)).

Landmarks also tend to have a unique visual appearance that leads to high discrimination values between different landmarks.

The automated classification of landmarks is based upon the photographing behaviour of users on large scale photo-sharing websites. Users tend to visit similar destinations and landmarks. When at these places, they also tend to take images of these landmarks from a small number of locations due to geographical constraints and the photogencity of an image from certain viewpoints. This leads to a huge overlap of visually similar images of popular landmarks. Based on this premise, this research takes advantage of this overlap by reducing the search space in a large scale dataset by clustering similar images thus creating more robust means of classifying an image using SVMs.

A large amount of previous work has been carried out in this research domain and although a relatively young research area, several solutions and methods have been suggested and implemented to address this research problem. These methods can be roughly classified into a number of types of approaches:

- Global Feature Based
- Local Image Feature Matching
- Geographical Based
- Inverted Visual Words
- Tree Based Approaches
- Model Based Approaches

Global Feature Based

The earliest landmark and scene classification systems were based on global image features such as colour and texture features as described in section 2.2.3.

Tackeuchi and Herbert [Takeuchi and Hebert, 1998] proposed an early landmark classification system based on three low-level image features; normalised red image histogram, Haralick texture features [Haralick et al., 1973] and an edge based histogram. They grouped similar images into a model and used different distance metrics to classify a test image against this model. They showed promising results, however, this approach was only evaluated using a very small set of images (68) and it is assumed that it would not scale to a much larger corpus.

Torralba et al. [Torralba et al., 2003] developed a helmet mounted mobile location classification system based on a wavelet based texture features using Gabor filters. This system could recognise 60 different locations such as a location on a specific street or a specific room within a building. The system could also classify objects within these locations based on a combination of content and contextual information using Bayesian probability. For example, an object would have a higher likelihood of being classified as a chair if the system had already classified the location as being in an office.

Yeh et al. [Yeh et al., 2004] developed a mobile based landmark classification engine that augmented landmark classification with web search. Their system compared images taken on a mobile device against a corpus of landmark images provided by a stock photography company (Corbis), and text based annotations that were associated with a successful match were used as queries to the Google search engine to retrieve information and images about that landmark. To perform the image matching they use a k-nearest neighbour approach based on two global image features (frequency information in the Fourier Spectrum and a wavelet based texture feature).

Local Image Feature Matching

A more advanced two stage classification approach was developed by Kosecka and Zhang [Zhang and Kosecka, 2007]. This approach utilised localised colour

histograms to prune a sample dataset before carrying out SIFT point to point matching. The results of experiments were promising with an accuracy result of 90.4% and a hit rate of 94.8% in the top 5 nearest neighbours. However, the dataset (ZuBuD dataset) used in this work was quite small and the spectrum of visual differences in building types is quite narrow. In addition to this two stage approach, the authors, improved upon it by training probabilistic based models in the SIFT matching phase which increased classification accuracy to 98.5%. Due to the small size of the dataset, it is not known how well this approach would perform with a large scale corpus; however it is assumed that as the size of the dataset increases, the accuracy of the approach would decrease significantly. In this work, a similar approach is implemented which improves upon the work of Kosecka and Zhang by ensuring that the size of the search space is kept to a minimum before interest point matching, thus ensuring a high classification accuracy.

Geographical Based

One of the earliest landmark classification systems combining content and spatial information is the 'EXTENT' system [Qamra and Chang, 2008] developed by Qamra and Chang. The Extent system combined content and context analysis to compare a test image against a dataset of sample landmark images. In the first stage of classification, GPS or Cell tower identification tags where available were used to prune the sample image dataset. In situations where GPS or Cell id information was unavailable, the system tried to infer some spatial information based on annotation information provided by humans (e.g. tag annotation contains phrase 'New York City'). Once spatial pruning had taken place, the system uses expensive point to point matching using SIFT image features. Rudimentary geometric constraint analysis was then carried out to certify a match. The system showed promising classification accuracy, however the matching process

employed was extremely slow. On average each test image required four hours to classify. Clearly this timeframe would not be suitable for a casual user, and certainly would not be suitable for a near real-time classification system.

Inverted Visual Words

Inverted Visual Words were first proposed for large scale image matching in [Sivic and Zisserman, 2003]. Using inverted files with visual word features is an efficient way to match a test image against large corpora. They provide a high level of discrimination, require a small heap memory footprint and provide a method to match large numbers of images quickly. Inverted visual words are described in more detail in Chapter 7.

Philbin et al. [Philbin et al., 2007] proposed an efficient method for landmark recognition consisting of an inverted visual word approach, followed by a fast spatial re-ranking procedure. They utilise large vocabulary sizes (k =10,000, 20,000, 50,000 and 1,000,000) and calculate inverted visual word features for each image from their corpora. Inverted features from test images are then compared against the index of inverted words using word frequencies as weights. Their approach is evaluated using a 3 different image corpora and they report a MAP score of .645 using a vocabulary size of one million.

Tree Based Approaches

Tree based approaches to matching local image features allow for accurate matching (usually approximate nearest neighbour approaches) in a very short timeframe. The first tree based approaches utilised K-d trees [Bentley, 1975], which are similar to binary trees but allow for the storage of local image features in k dimensional space.

A landmark approach was developed at the information retrieval company Google that made use of graph data structures. Zheng et al. [Yan-Tao Zheng, 2009]

used a large-scale parallel computing system to extract landmark images from a collection of over 21 million community contributed images. This community data was augmented with web images gathered using a novel approach based on the extraction of possible landmark names from a travel guide website (wikitravel.com), and subsequent web search for images based on these landmark names. These images are organised into a graph structure using interest point matching with nodes representing images and edges representing matched regions between images. Once the graph structure is created and landmark images are identified, a K-d tree structure is created to index their local features and test images are matched against these images in real-time.

Model Based Approaches

Another approach to landmark recognition is to build a model constructed from a number of visually similar images, and then utilising different classification methods (for example, machine learning classification techniques such as neural networks or support vector machines), to classify the presence of a landmark within an image based on this model.

Li et al. [Li et al., 2009b] extracted from a large scale image corpus, a collection of images containing 500 of the worlds most photographed landmarks. Each of these landmark clusters were based upon peaks in photo distribution when searching the Flickr API using geographical coordinates as inputs. The top 500 peaks in this search were chosen as the top 500 landmarks worldwide.

A single multiclass SVM model was then trained to recognise each of these 500 landmarks using visual bag of word features as the inputs with a vocabulary size of 20,000. They also combine these visual features with textual information associated with each image. They reported a classification accuracy of just over 45% when classifying across 500 landmarks. However, they also experimented with training models to recognise smaller numbers of these landmarks and when

this number was reduced to the top 10 landmarks, the classification accuracy was increased to over 80%. The techniques used in the work described in this thesis are similar to that of Li et al. [Li et al., 2009b] in that multi-class SVMs are utilised for landmark classification, however, the aim is to classify a much larger spectrum of landmarks (popular and non-popular), even those that might not have a high generality, which is a more challenging task. In this work, it is also intended to improve upon the results of their SVM classification accuracy, using a more efficient manner in which to cluster training sets of data, including taking effects of affine variation into account along with adding overlap visually similar images to different clusters to aid robustness.

Popescu et al. [Popescu and Mollic, 2009] use a k-Nearest Neighbour approach to differentiating between clusters of landmark images. They use a dataset consisting of a collection of the most commonly photographed landmarks in the world, each landmark image cluster is created by querying Flickr and Panoramio with the name of a landmark and populating the returned results into the cluster. To classify an image they use the cumulative scores of the distances between the top 5 nearest neighbours for each landmark cluster. This approach however, only accounts for the classification of very popular landmarks and makes no provision for landmarks that might not be densely represented in a community dataset. In this thesis, the aim to improve upon Popescu's approach firstly by reducing the amount of noisy images in each image cluster through the use of visually similar image clusters as opposed to semantically similar and additionally providing a means to classify uncommon viewpoints of landmarks within community corpora.

Several other model based approaches that have been implemented, and include the novel use of a scene map model to represent a scene in community data imagery. Avrithis et al. [Avrithis et al., 2010] cluster Flickr images based on geographical data followed by a visual clustering process to group sets of visually

similar images. Once this clustering has taken place they calculate homographies between images within a cluster using the RANSAC algorithm and align all features. They then construct a 2d spatial map (which they call a scene map) of all features in that scene extracted from all the different viewpoints. These scene maps are the treated as training images, and test images are then compared against each of these scene maps using the visual bag of words approach with inverted index files. They compare this approach against a baseline visual word method and illustrate significant improvements in classification accuracy.

Another novel model based technique is the use of 3d point clouds to extract only interest points from the sections of an image where the landmark is located. Xiao et al. [Xiao et al., 2010] developed a technique to improve landmark classification based on creating a model of a scene using 3d reconstruction methods, specifically a structure from motion technique [Snavely et al., 2006]. Using these 3d reconstruction models they calculate the regions of an image that contain the landmark by projecting the 3d points to 2d space. Image features are extracted from these landmark regions and test images are matched against the landmark region features using a kd-tree data structure. While the results of this work are quite promising, the dataset used in the experiments was quite small, containing only 6 landmarks. Additionally also the processing time required is quite expensive.

2.8 Summary

In this chapter, a brief history of the techniques proposed and used in the field of image retrieval are introduced. The background concerning the computer vision technologies used in this thesis was then described and motivations behind the usage of these features was presented. The chapter was concluded with a description of several of the alternative methods proposed in the literature to

solve the problem of automated landmark recognition. The main aim of this work is to improve upon many of these previously suggested approaches in three main categories: recognition precision, classification time and memory requirements.

The framework proposed in this thesis builds upon much of the previous work in the field, taking advantage of techniques that have been shown to work well while disregarding others and improving upon many of the inefficient methods in the literature. Many of the alternative techniques in the literature are based on the indexing of interest point features which require these features to be stored in a heap memory structure. This method places a limit on the number of images that can be included in a training corpus. The framework proposed in this thesis improves upon this by ensuring that there is a static memory requirement regardless of corpus size.

Additionally, the framework proposed in this thesis will improve upon many of the interest point matching schemes previously suggested in the literature by first effectively reducing the search space to a small subset before interest point matching is carried out. This search space reduction allows for the recognition of landmarks in real time even with a large scale training corpus.

Chapter 3

Community Contributed Datasets

3.1 Introduction

The term 'Web 2.0' was first coined by Tim O'Reilly at the O'Reilly Media Web 2.0 conference in 2004. Web 2.0 refers to a fundamental change in how the internet functions, and how average users are able to create and share content online. The phrase is now used to describe methodologies and technologies that allow and facilitate the sharing of information between internet users. One of the main paradigms of 'Web 2.0' is the emergence of online user generated content. Before the arrival of 'Web 2.0' most information online was generated by a small number of people, usually from a small spectrum of society such as IT graduates, technology enthusiasts and large businesses. Most average users did not have the time or expertise to develop websites and populate them with content due to the complexity involved. Most utilised the internet as a means of retrieving information but they had little opportunity to publish and edit their information online.

The term 'Web 2.0' has been quite controversial as there is no strict definition of the phrase. In a blog entry in September 2005 Tim O'Reilly outlined his understanding of the phrase, saying 'like many important concepts, Web 2.0

doesn't have a hard boundary, but rather, a gravitational core. You can visualise Web 2.0 as a set of principals and practices that tie together a veritable solar system of sites that demonstrate some or all of those principals at a varying distance from that core [O'Reilly, 2005].

Irrespective of how the exact definition of 'Web 2.0' is interpreted, several technologies have been developed in the last decade that follow the methodologies of O'Reilly's 'gravitational core'. Since the O'Reilly media Web conference in 2004, thousands of websites and online applications have emerged that follow the core principals of the 'Web 2.0' ideal. Many new methods have been developed to distribute information in different ways and more efficiently across the web. Large numbers of web users are now not only using the web for the retrieval of information but actively creating and distributing information online using Web 2.0 technologies. Some of the new tools and methods that have emerged and are referred to as Web 2.0 applications are:

- **Wikis.** Wikis are online collaborative websites that allow users to exchange resources including text and multimedia. The main feature of wikis is that they are collaborative. An entry in a wiki can be created by many users. Once data is added to a wiki, all other users of the wiki can edit or delete that data. One of the main aims behind wikis is that any user can add or edit content with ease and without the need for specialist technical knowledge. One commonly used wiki is the online encyclopaedia 'Wikipedia' [Wikipedia, 2001], which allows users to upload and edit encyclopaedic entries and currently over 300,000 users contribute to Wikipedia each month (June 2011).

- **Blogs.** Blogs are another application of 'Web 2.0' that have become extremely popular in recent years. A blog is a webpage that acts as an online journal for the owner. A blog consists of regular journal entries that contain the

creator's thoughts, ideas, multimedia entries and sometimes web links. Readers of a blog are able to leave comments on each entry and this can soon lead to large discussions building up quickly involving many users. Many blogs can be created using software packages that do not require technical expertise to use them efficiently. Some commonly used examples of weblog software include WordPress [WordPress, 2003] and Movable Type [MovableType, 2001].

- **Social Networking.** Social networking sites have become very popular in recent years with many of the sites becoming household names such as MySpace [MySpace, 2003] and Facebook [Facebook, 2004]. The social network Facebook alone has over 750 million active users (June 2011). An online social network is a website that allows a user to create an account, usually called a profile. A user's profile can then be linked to their friend's accounts and others within their social circle. A user can populate their profile with information about themselves, such as images and details describing hobbies and interests. A user can generally make their profile public which means that it can be viewed by all members of the social network, or private which means that it can be viewed only by that user's friends. One common feature of social networks is the ability of users to maintain a blog on their profile page for others to read.

- **Social Bookmarking.** Social bookmarking is an application where users can share links to webpages with other people. A social bookmarking site is a content management system that allows users to upload and store bookmarks that they find interesting or useful. A user can then share these links with the wider community or just with other users within their social circle. Social bookmarking originated from a desire for organisations to share information between members mainly within academia within a short

timeframe. It has since grown to become very popular worldwide with many commonly used social bookmarking sites being used by millions of people regularly such as Digg.com (45 million monthly vistors - July 2010) [Digg, 2004] and Reddit.com (16.5 million monthly visitors - July 2010) [Reddit, 2005].

- **Video Sharing.** Another popular facet of Web 2.0 is online video sharing. Video sharing websites have grown in popularity hugely in the last 5 years. There are several reasons for this, including the reduction in cost of camcorders and video creation devices and the constant increase in internet connection speeds. Several years ago it would have taken a user many hours to upload or download a large video file to or from a webserver, whereas today with a high speed broadband internet connection the same file might take minutes or even seconds to transfer. This has led to the creation of websites that allow users to upload and store videos on webservers that can be shared and viewed by other users of the site. Several video sharing websites also contain a social network aspect, allowing users to create profiles and playlists and comment on other user's videos. The most commonly used video sharing website today is YouTube [YouTube, 2005]. Such is it's popularity, that in August 2010 Alexa internet, a subsidiary company of Amazon that tracks of internet traffic, reported that YouTube was the third most visited website in the world [Alexa, 2010].

- **Photo Sharing.** Digital cameras have become more accessible to people over the last few years and as a result more and more digital imagery is being created. Image capture and storage in particular has undergone many large changes in the last decade. In the past, users tended to keep personal photo collections private and store them in photo albums or digitally on personal

computers at home. In recent years however, users have now started to share their personal photo collections online with the world.

Many websites have been created in recent years that allow users to upload imagery and store it, where it can be viewed by others. Several of these websites follow the guidelines of 'Web 2.0' and contain social network aspects. Many of the online photo sharing websites allow for the creation of profiles and groups. Many sites allow for the addition of tags (short textual annotations) to describe the content of an image, which allows for the browsing of photos by different categories. The most commonly used photo-sharing website in use today is Flickr [Flickr, 2004]. A screenshot of the Flickr interface is presented in Figure 3.1.

3.1.1 Community Contributed Data

The large increase in the amount of users creating and distributing data online has led to the creation of multimedia datasets of unprecedented scale. There are huge amounts of multimedia being stored on servers worldwide that have been created,modified and distributed by hundreds of millions of people. Many new research genres have been created based on how to organise and retrieve this data.

In this work, the focus is on utilising large collections of images that have been created by online communities of people. Specifically, the focus is on geo-tagged imagery, which is imagery containing some geographical data describing the location where the image was taken.

3.1.2 Geo-Tagging and Global Positioning Systems

The 'Global Positioning System' (GPS) is a satellite based system that provides accurate time and location data anywhere on the planet. The system consists of

Figure 3.1: An illustration of the Flickr interface displaying an image. Marked in the illustration are the different types of metadata associated with the image, along with various social network information.

24 satellites that orbit the earth in three separate orbital planes. Each satellite broadcasts information consisting of orbital information and time information. A GPS receiver with line of sight to at least four satellites can work out the longitude and latitude of the receiver based on these broadcasts. The GPS system is very accurate and can predict a receiver's geographical location to within a radius of ten metres.

One method for storing descriptive metadata about an image at capture time is to store it in a header file. One of the most commonly used standards today is

the Exchangeable Image File Format (EXIF). The EXIF standard is now supported by most of the large digital camera manufacturers and it has also been adopted by many smart phone producers. The EXIF standard allows data to be stored in the header of an image providing contextual information describing basic camera settings (time of image capture, aperture, focus length, etc..). The standard also supports geographical information in the form of longitude/latitude coordinates.

Image geo-tagging refers to the process of associating a geographical position (ideally the location where the image was taken) with an image. This process may be carried out automatically on the image capturing device or manually by the user. Many high-end digital cameras and many new smartphones come equipped with GPS receivers, thus allowing the automated geo-tagging of images with a high degree of accuracy. These devices can embed the GPS position directly into the EXIF header of an image, which can then be extracted easily by software at a later stage. If the camera does not support automatic geotagging it is still possible to insert longitude/latitude coordinates after image capture using third party software and a GPS device.

In the absence of a GPS device or a GPS enabled camera, photographs can still be associated with geographic co-ordinates by scrolling, zooming into and selecting a location on a map [Panoramio, 2005][Toyama et al., 2003]. This process is called 'Geo-Tagging' and is carried out manually by a human annotator.

Geographical information has become very important in the field of information retrieval. Sanderson and Kohler [Sanderson and Kohler, 2005] analysed large numbers of queries sent to the then popular search engine Excite with the aim of measuring how many contained geographical locations. Their work showed that almost 20% of these web queries referred to a geographical location. Another analysis of over 36 million AOL queries, revealed that 13% of the queries submitted to the system referenced a specific place or landmark [Gan et al., 2008].

3.2 Creation of Geo-Tagged Datasets

At present there is no standard large scale geo-tagged photo dataset used by the wider research community for analysing the effectiveness of landmark classification techniques. As part of this work it was necessary to create an appropriate dataset with which to carry out experiments to test the hypotheses outlined in Chapter 1. Several small scale landmark datasets are available such as the Paris landmark dataset and the Oxford landmark dataset, both released by the Visual Geometry group at Oxford University and used in the papers [Philbin et al., 2008] and [Philbin et al., 2007]. These datasets are quite small, however. They do not contain geographical information and have been used to test image recognition techniques only.

Several other groups have created datasets for use in landmark categorisation and classification. Crandall et al. [Crandall et al., 2009] created a dataset consisting of 35 million Flickr images with the aim of organising them effectively and revealing interesting properties about world cities and commonly photographed landmarks. As part of their work they analysed the textual tags that accompany Flickr images along with some quantised SIFT feature vectors [Lowe, 2004]. They combined these features to group large numbers of images into clusters based on location and landmark. Lists of the most commonly photographed landmarks in the world were created and representative images of the most commonly photographed landmarks contained within a city were generated for many world cities. This dataset has not been publicly released by the authors, due to the disk space required to store a dataset of this size, the time and bandwidth required to transfer this dataset online and perhaps some licensing issues with redistributing data created by thousands of people. Additionally, as this was not a classification task there is no test collection of images with which to test the accuracy of the landmark clustering process.

Two large online photo websites that specialise in storing and displaying geo-tagged imagery are Panoramio and Flickr. Both of these websites allow access to sub-sets of their collections through web based application programming interfaces (API).

Panoramio

The Panoramio website [Panoramio, 2005] was launched in 2005 as a photo sharing website that specialises with images that have accompanying geo-location information. It was one of the first image sharing websites that dealt solely with geo-tagged images. The site quickly became popular after it was launched and within two years contained over 5 million images. When a user uploads an image to Panoramio they can provide a weak annotation of the image by creating textual tags describing the image.

Panoramio have developed a publicly available API that can be used to access and display their content. Using this API, all images within the Panoramio dataset can be accessed and download links can be requested. A search to the API consists of sending bounding box coordinates, and the service will return data describing all images located within that region. The site was bought by Google in July 2007 and the images stored on the site were integrated into the popular Google Earth application. There are some restriction issues with the usage rights of Panoramio's data, therefore the main corpus used in this work was collected from another source, Flickr.

Flickr

Flickr [Flickr, 2004] started off its life as a mini application that was developed by an engineer working with the online game firm Ludicorp. This allowed users to take pictures of their current progress in a game and upload the screenshots to a web page. One of the firms founders, Caterina Fake, saw potential in this

application and decided to cancel game development to solely concentrate on a web-based photo sharing system. This system was launched in February 2004 and became known as Flickr.

Flickr was purchased by Yahoo in 2005 and has now grown to become one of the most popular and largest photo sharing websites in the world. Over 50 million people are now registered Flickr users worldwide. The website has an average 5.3 million visitors per day [TechCrunch, 2010]. There are now over 4 billion user uploaded images stored on the Flickr servers. Flickr also contains the largest collection of geo-tagged images in the world. Over 130 million images have been uploaded to flickr and either automatically geo-tagged by GPS enabled camera devices or manually geo-tagged by users.

Along with being a large photo repository, Flickr is also a social network. Users can create accounts that allow them to add and contact friends, join social groups and limit viewing of their images to certain people within their social circle. Users can also comment on other user's photos, add others photos to their list of favorites and suggest groups for the photos to be associated with.

3.2.1 Creating a Geo-Referenced Landmark Dataset

There were several challenges involved with collecting a landmark image dataset based on publicly provided geo-tags and annotations. One of the biggest challenges is trying to differentiate between images containing a landmark as the main subject within an image, and images taken within the locality of a landmark but not actually containing the landmark itself. It is quite common that Flickr users will tag an image of an event that has taken place in the locality of a landmark with the name of the landmark.

The term landmark can be quite subjective. Several types of building, geographical features, and monuments could be considered to be landmarks. These

landmarks can take many different shapes, colours and sizes, and due to these differences it is very difficult to automatically classify an image as containing a landmark using computer vision techniques alone.

One set of features that can help to distinguish between Flickr images containing landmarks and images depicting events are textual features. The semantic textual annotations that accompany Flickr images can be used to filter out event-based images and find landmark images from a dataset. Abbasi et al. [Abbasi et al., 2009] developed a technique for finding landmark images from large Flickr datasets using a binary SVM classifier to classify whether an image contained a landmark based on their associated text annotations. This approach used collections of known landmark imagery and used their tags to train an SVM model, while using tags from arbitrary groups of photos labeled with generic tags such as 'birds' and 'airplanes' as negative inputs.

Ahern et al. created a data visualisation tool to display landmark images and landmark tags within small geographical areas [Ahern et al., 2007] based on image tagging habits of Flickr users. They utilise a system based on term frequency and inverse document frequency scoring of the tags. For each small geographical region the application displays high scoring tags along with images associated with these tags, usually depicting landmarks and places of interest within the region.

3.2.2 Harvesting a Geo-Referenced Landmark Dataset

For this investigation it was desired to create a dataset of geo-tagged imagery that covered an entire metropolitan region of a large city. As geo-tags provide an efficient means to filter any large dataset of images worldwide, it is assumed that any approaches to landmark classification using the image data from one city could be replicated on a larger dataset containing data from many cities, or

possibly even a very large image corpus that covers the entire world. The accuracy and time required to carry out the framework described in this work would not be hindered by enlarging the dataset to include other regions, as spatial filtering techniques could quickly prune out all images outside of a query image's region.

The city of Paris was chosen for this work. This is mainly because Paris contains a large number of objects and locations that could be considered landmarks, and in certain regions within the city there is a high distribution of landmarks. Additionally, the Parisian region is one of the most densely populated regions that is represented on Flickr with regards to geo-tagged photographs (490,000 in Paris region (June 2011)). A large training collection of images was harvested from Flickr for this purpose along with a test collection to evaluate the framework, all located within the Parisian area. An example of the wide availability of a Parisian landmark in the Flickr archives is presented in Figure 3.2.

The training collection of geo-tagged images was harvested using the publicly available Flickr API. When using the Flickr API, users can provide a text query which is used by the Flickr system to return images relevant to that query. To return possible landmark images, the Flickr system was queried with a list of generic words that might indicate a landmark is present in an image, such as 'landmark', 'church', 'bridge', 'building', 'facade' etc..

In this work, the approach that was proposed to filter out non-landmark imagery from harvested images is based on the presence of certain tags that might indicate that an image is depicting an event, people or an object, rather than a landmark. A list of tags was created, each of which occurs frequently in sets of images that do not contain landmarks and do not tend to occur in sets of images that do contain landmarks. This list of tags are labelled as candidate tags and images containing one of these candidate tags were then filtered out from the data set.

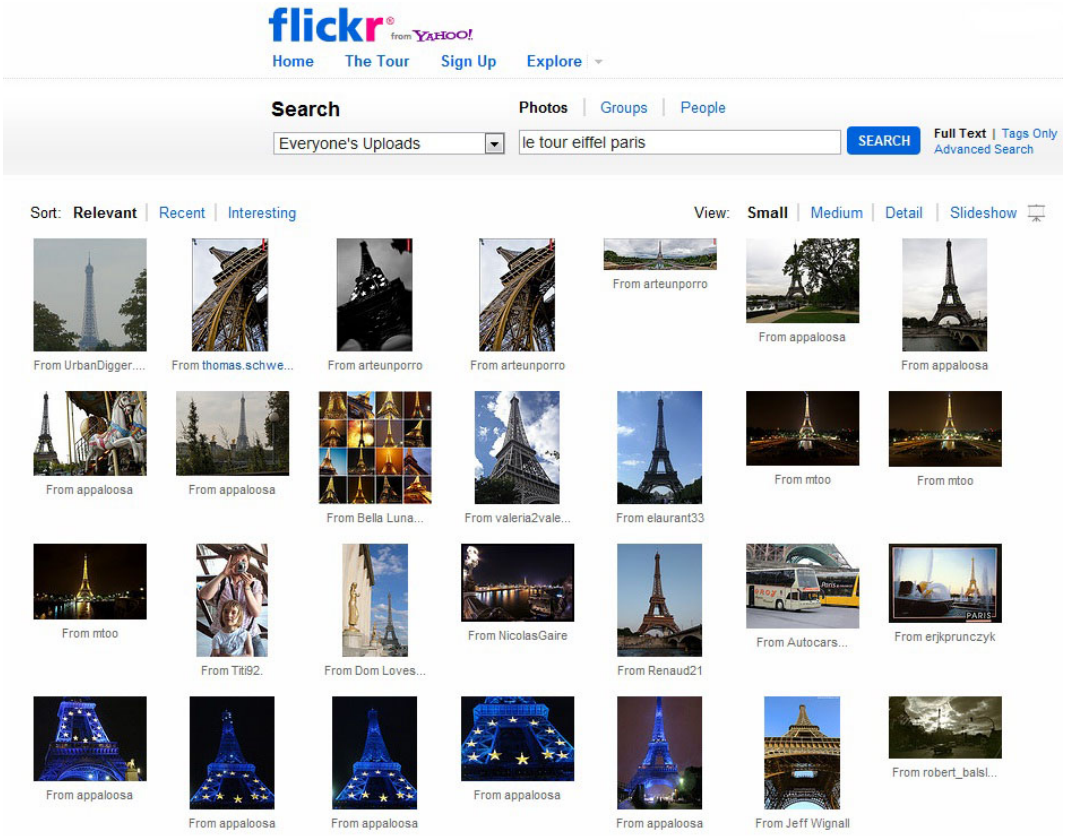

Figure 3.2: An illustration of the Flickr online interface. Displayed are the top results returned by the Flickr system using the query string "le tour eiffel paris".

To build a list of candidate tags, an image set collected from Flickr consisting of 1000 images was manually inspected and classified as containing a large landmark. This set was labelled as S_1. A further set of 1000 images that did not contain a large landmark, but rather depicted an event or different types of objects, people, and animals was also collected and denoted S_2.

For the set S_1 a list of tags was created denoted as T_1, containing all tags that were associated with the images contained in S_1. Another list of tags T_2 was created containing all the tags associated with images in S_2. All tags contained in $T_2 \backslash T_1$ were considered possible candidate tags, however the presence of a tag in $T_2 \backslash T_1$ alone is not enough to indicate that the tag would suggest a non-landmark image. It was decided therefore, to selected the tags that occurred the highest number of times in T_2 but not T_1. The final set of candidate tags was selected based

Table 3.1: Most frequent filtered tags

Parade	Marathon	People	Wedding	Concert
Bar	Dinner	Friends	Cat	Dog
Flower	Pride	Gay	Party	Sport
Game	Graffiti	Kiss	Love	Match
Demonstration	Band	Football	Reception	Springbreak

on the tag frequency of each possible candidate tag from $T_2 \backslash T_1$. The frequency was calculated using the following formula:

$$tf_i = \frac{t_i}{|T_2 \backslash T_1|}$$

where t_i is the number of occurrences of the tag i in the list $T_2 \backslash T_1$. If the term frequency was above a threshold of .005 (roughly translating to a frequency of 10), the tag was marked as a candidate tag. Some of the most frequent candidate tags are shown in Table 3.1.

The tags from all images downloaded from Flickr were examined, and any image containing one of these candidate tags was filtered out. In total from over 200,000 images downloaded from Flickr in the Paris region, over 100,000 were filtered out using this approach, leaving a final training corpus consisting of just over 90,000 images. From informal empirical inspection this tag filtering approach seems to work quite well, with the vast majority of images in our dataset depicting a place or landmark.

3.2.3 Training Collections

After this filtering process the main training corpus used in this work consisted of 90,968 images containing a landmark as the main subject of the image. A randomly selected subset of this corpus is presented in Figure 3.3. Each image in the collection contains a set of context information. For each image there is a set of photo context information:

- **Image ID.** Each image in the dataset is represented by a unique identification number that is represented as an integer.
- **User ID.** Flickr identifies people by using a unique user name for each user of the system.
- **Image Title.** When a user uploads an image to Flickr they are given an opportunity to suggest a title name for that image. This title is optional and many users choose to leave it blank, in which case the Flickr system will consider the file name of the image as the title. This can lead to many generic title names that are created by the device used to capture the image.
- **Location Information.** All images in the dataset contain spatial information, describing the location where the image was taken. This information comes in the form of longitude and latitude coordinates.
- **Textual Tags.** When uploading an image to Flickr, users are encouraged to suggest a set of words (called tags) describing semantic information about an image, which are then organised as a list of words.
- **Licence Information.** Licence information is available for all images within the dataset. The license information is represented by an integer, which corresponds to a license type outlined on the Flickr website.

3.2.4 Test Collections

Flickr was also farmed for a test collection of images that consisted of 1000 random landmark photos in Paris. The test collection was created in the same manner as the training collection using the same list of candidate tags to filter out non-landmark images. The images were searched using the generic tag 'landmark', which combined with analysing the associated tag information and removing

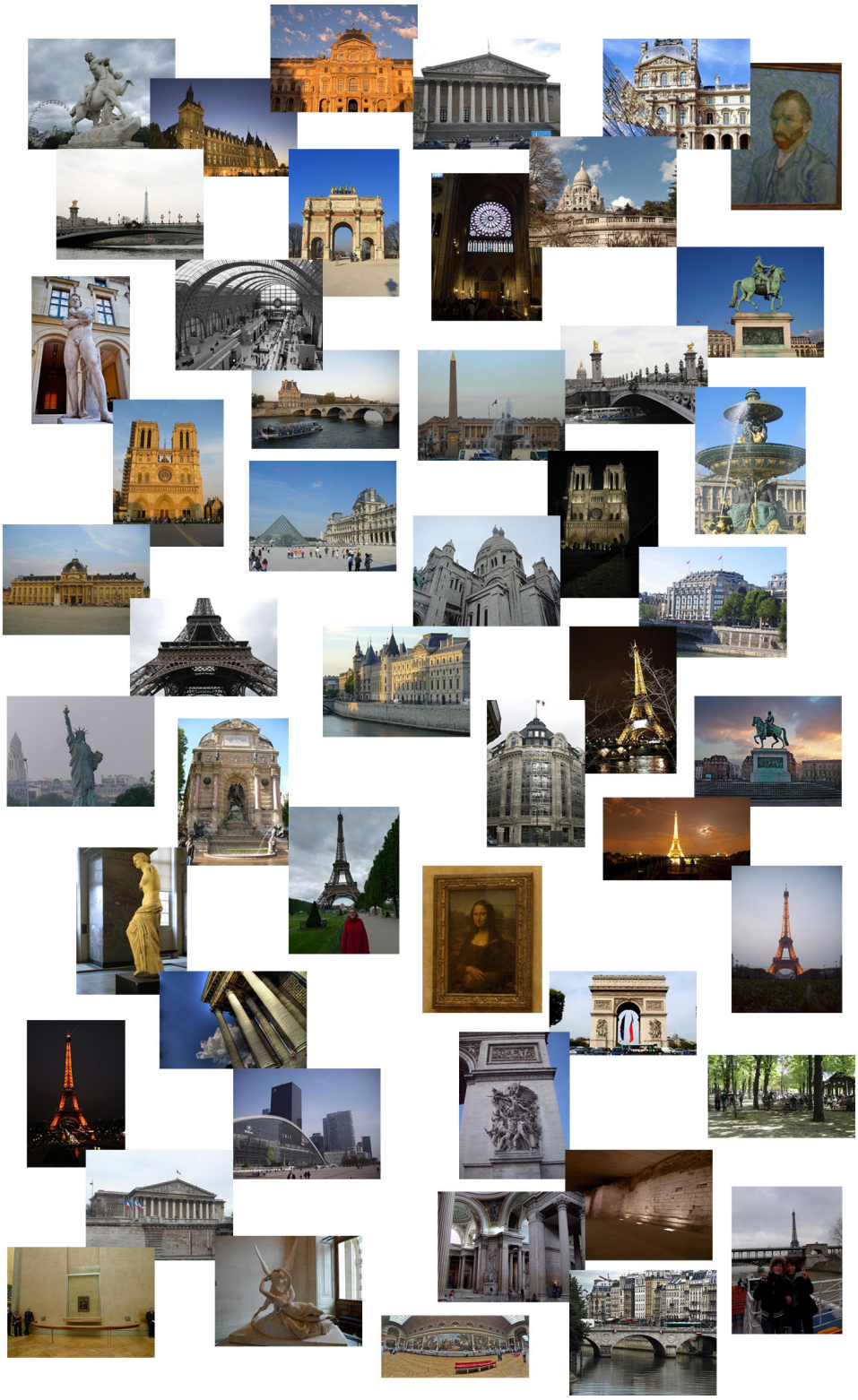

Figure 3.3: A random subset of images from the training corpus used in this work.

images containing any of the candidate tags, seems to work accurately from empirical inspection.

To differentiate the training collection from the test collection, Flickr was searched using an option to rank the images returned by upload date. Only images with an upload date later than the latest upload date from any image within the training collection were considered. As with the training collection, each image in the test collection contained metadata in the format: image id, user id, image title, location information, textual tags and license information. For the purposes of the evaluations carried out in this work, the textual tags and image title information was dismissed and unused.

3.3 Analysing Community Contributed Metadata

In recent years there has been a dramatic increase in the volume of community contributed resources online. The sheer size of these resources has created many new research opportunities. Community contributed data sets are undoubtedly useful resources for research purposes, particularly in the collection of large amounts of training and testing data for experimentation. There are several drawbacks however, in the use of this data, in particular with community contributed imagery and metadata.

3.3.1 Analysis of Geographical Information

Flickr provides an interactive map interface to users while uploading an image to geo-tag it. A user can pan and zoom to the location where they believe the image was taken. The tagging system will then associate this location with the image in the form of longitude and latitude coordinates. Some users will pan and zoom to a very accurate location, while some users will not zoom enough to create accurate geographical information. It is also possible that some users will not be

aware of the exact location where the image was taken, and incorrectly pan and zoom to a different location altogether. Another potential issue is that humans tend not to think of location in terms of latitude and longitude coordinates, but rather in vague spatial relations such as 'near O'Connell Street, Dublin' and might possibly tag a image with a geo-location that belongs to a nearby landmark, street or region, rather than that of the image itself. As spatial information plays an important role in the overall framework outlined in this thesis, it is vital to know to what extent these inconsistencies affect the accuracy of the metadata.

Several research groups have carried out analysis on the accuracy of geo-tags in community image collections, in particular using Flickr datasets. Girardin and Blat [Girardin and Blat, 2007], carried out a study on location information granularity using Flickr geo-tags along with other metadata. When users perform geo-tagging on the Flickr system they zoom to a specific location. The Flickr system automatically assigns a zoom level attribute to a geo-tag based on the zoom level that the user geo-tags at. These zoom attributes were analysed in a collection of 1.6 million Flickr images taken in 12 cities around the world. They noted that peaks seemed to appear in graphs of geo-tags at zoom levels 12 and 16. Zoom level 12 is defined as a general city level, while zoom level 16 is determined to be at street level. The authors not only wanted to discover the zoom levels users tended to geo-tag an image, but also if a user's familiarity with an area effects the accuracy of a geo-tag. The time data associated with an image was noted and any user that had uploaded photos in a city with time stamps more than two months apart was deemed to have familiarity with a city. This research seemed to suggest that a user's familiarity with a region did not effect the accuracy of the geo-tag. It must be noted that this work did not approximate the actual locations of images, but only the zoom levels of the map interface provided by Flickr, therefore, there is no check to see if the geo-tags are accurate or not. In this work, the aim is to

improve upon the work of Girardin et al, by analysing the accuracy of the geo-tags over the level of zoom that users commonly use when geo-tagging an image.

One of the most detailed analyses of Flickr geo-tag accuracies was carried out by Hollenstein as part of her research for her masters thesis at the University of Zurich [Hollenstein, 2008]. In this work, just under 10,000 images taken within London were gathered that had a tag 'hydepark', representing the public park. A bounding box surrounding the park was created and the geo-tags associated with the images examined to check if they were located within this bounding box. Over 86% of the geo-tags were found to be located within this bounding box, with varying levels of granularity. Although the significant majority of these geo-tags were found to be located within the bounding box, this information still only provides a rough indication of the accuracy of geo-tags as no experimentation was carried out to inspect whether or not the tagging was accurately associated with the image by the user.

It remains quite difficult and time consuming to garner a precise measurement of the level of accuracy of each individual geo-tag, since only the person who captured the image can be sure to a high degree of accuracy where they were located at the time of image capture. Based on this premise, it is assumed that some element of manual inspection and a fundamental level of local geographical knowledge is required to measure precisely the level of accuracy contained in this metadata. Inspired by the work of Hollenstein, and to address the issue of manual inspection, detailed manual analysis was carried out on a subset of the images contained within the Paris dataset to provide a reliable and accurate measurement of geo-tag precision. A subset comprising of 673 landmark images from the Paris training set were selected to be analysed. Based on local knowledge of the region, each of these images was estimated to have been photographed within very close proximity(approximately 100 metres) of four different landmarks in Paris (Paris Opera House, Arc De Triomphe, Louvre Pyramid and Pont Neuf Bridge).

The actual geographical centre point of each of these landmarks was noted, and a bounding box with side lengths of 200 metres was created. Each bounding box was centred around it's associated landmark centre point. The geo-tags of each of these 673 images were examined, and for each one the distance between the geo-tag and the associated bounding box was calculated to measure the accuracy of each geo-tag.

To calculate distances between two geographical locations, it is not sufficient to use Euclidean geometry distance formulas such as the Euclidean or Mahalanobis distance measures. This is because the Earth is a spherical object and the distance between two points on a sphere must be calculated using spherical geometry. To calculate distances between a geo-tag and a geographical bounding box the Haversine formula is used, which can be described in pseudocode in 6 steps as:

1. R = Radius of the Earth (6371km)
2. $dLat = latitude_2 - latitude_1$
3. $dLong = longitude_2 - longitude_1$
4. $a = sin^2(\frac{dLat}{2}) + cos(latitude_1).cos(latitude_2).sin^2(\frac{dLong}{s})$
5. $c = 2.atan2(\sqrt{a}, \sqrt{(1-a)})$
6. Distance = R $*$ c

The results of this analysis are quite interesting (presented in Table 3.2 and illustrated in Figures 3.4, 3.5, 3.6 and 3.7), in that they indicate that manually created geo-tags are accurate to within a certain radius. These results show that the majority of geo-tags that were examined are accurate to within 200 metres (over 80%). This is not as accurate as a modern, high end GPS receiver (generally accurate to within 10 metres, depending on the strength of the connection and line of sight), but should be accurate enough to allow for efficient filtering of

Distance(Metres)	50	100	200	250	500	1000	2000	>2000	**Total**
No. of Geo-Tags	372	506	545	552	578	599	625	673	673
% of Geo-Tags	55.2	75.1	80.9	82	85.8	89	92.8	100	100

Table 3.2: Results describing the number of correct geo-tags for each spatial radius, along with the percentage of correct geo-tags from the subset of those examined

unwanted images while clustering or classifying imagery using community data. It must be noted that a number of these images might have geo-tags that were created automatically (using a GPS enabled device) and not entered manually by the uploader. The results of the analysis are presented in Table 3.2.

3.3.2 Analysis of Human Defined Captions and Tags

Flickr supports annotation while an image is being uploaded to the site. Users can provide a title for an image along with a set of short tags to describe the semantic content of an image or extra contextual information. These text tags are freely entered by a user and do not have to adhere to any set of rules. These annotations are not part of any ontology or categorisation process, and therefore large inaccuracies can occur. Many users will enter tags that are heterogenous, as their interpretation of the semantic content within an image might differ from another users interpretation. An average Flickr user will generally not be concerned with how these tags might aid image retrieval and often will not spend the time required to create rich and accurate tags. This can lead to ambiguous and vague annotations that are not suitable to enable effective text-based image retrieval.

Several research groups in the past have analysed the usefulness of community contributed metadata for image retrieval tasks, in particular the relevance to their associated image, of textual tags provided by users uploading image to Flickr.

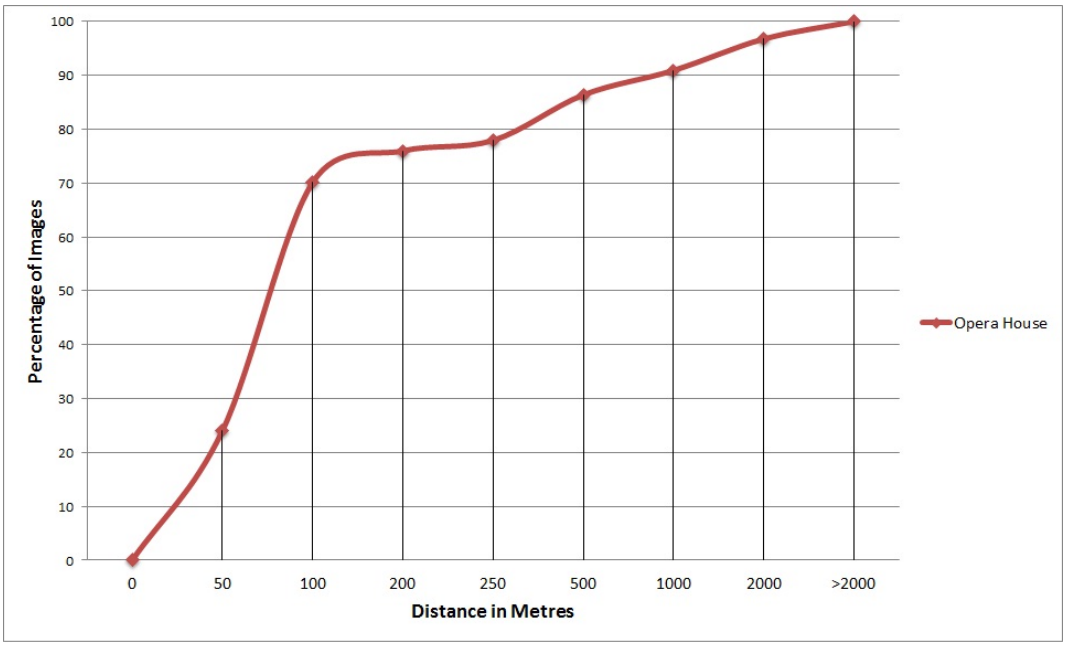

Figure 3.4: A graph illustrating the accuracy of the geo-tags for all images in the dataset taken of the facade of the Paris Opera House. As can be seen in the illustration, over 75% of all images are accurate to within 200 metres, with only 3% of images inaccurate by over 2 kilometres.

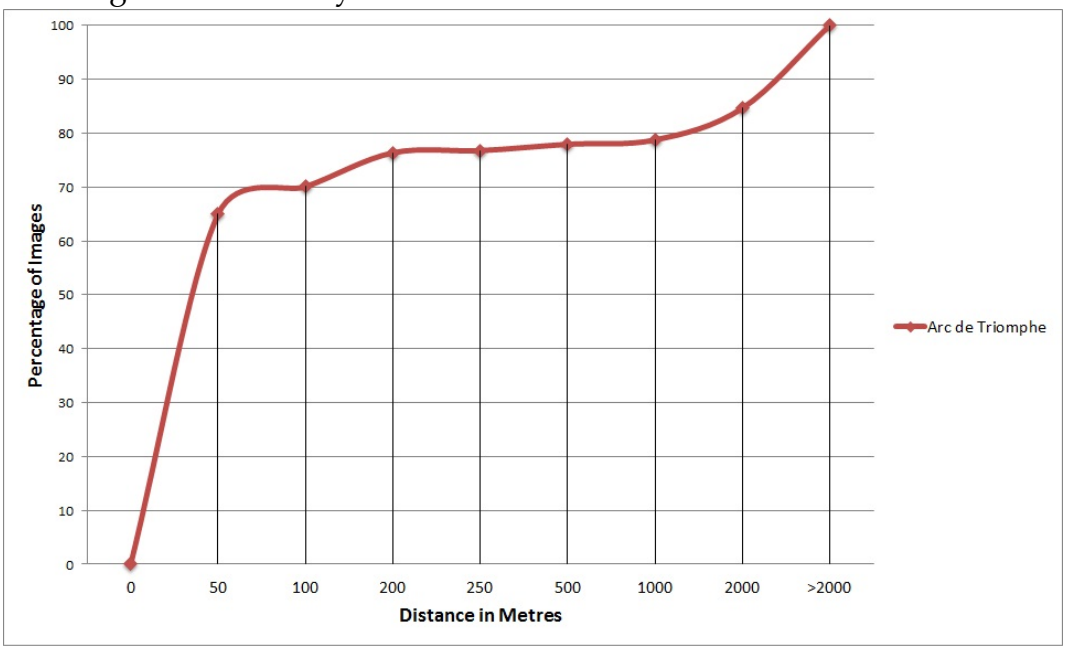

Figure 3.5: A graph illustrating the accuracy of the geo-tags for the images in the dataset taken of the Arc de Triomphe from nearby. Over 75% of all images are accurate to within 200 metres.

In their work on Flickr tag recommendation Sigurbjournsson and Van Zwol [Sigurbornsson and van Zwol, 2008] analysed tags associated with over 52 mil-

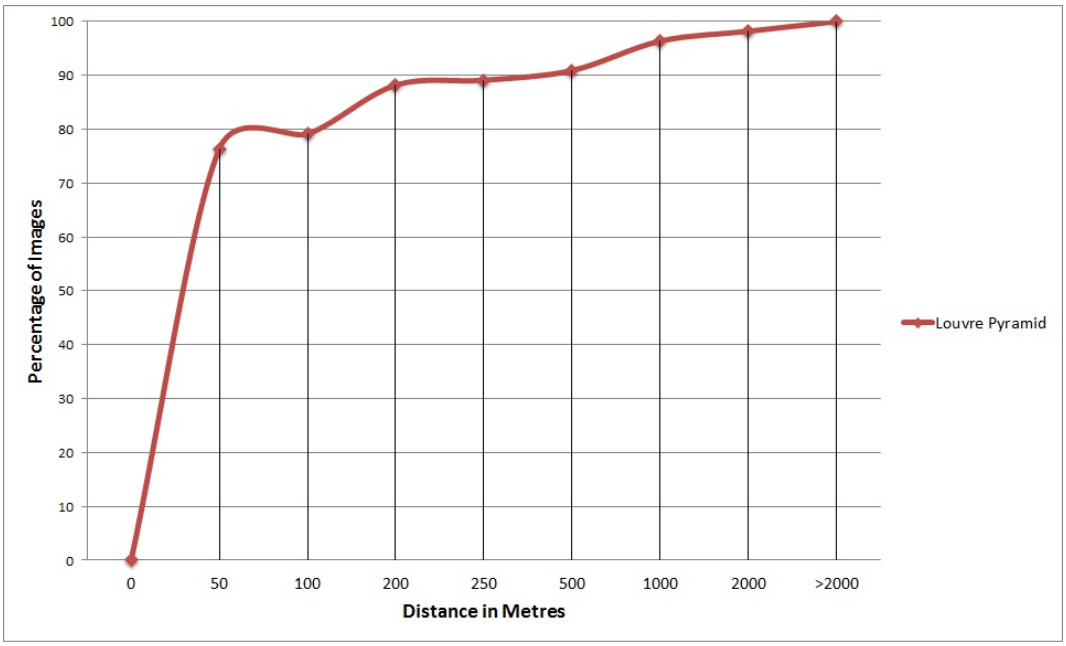

Figure 3.6: A graph illustrating the accuracy of the geo-tags for all images in the dataset taken of the Lourve Pyramid. As can be seen in the illustration, over 85% of all images are accurate to within 200 metres, with only 3% of images inaccurate by over 2 kilometres.

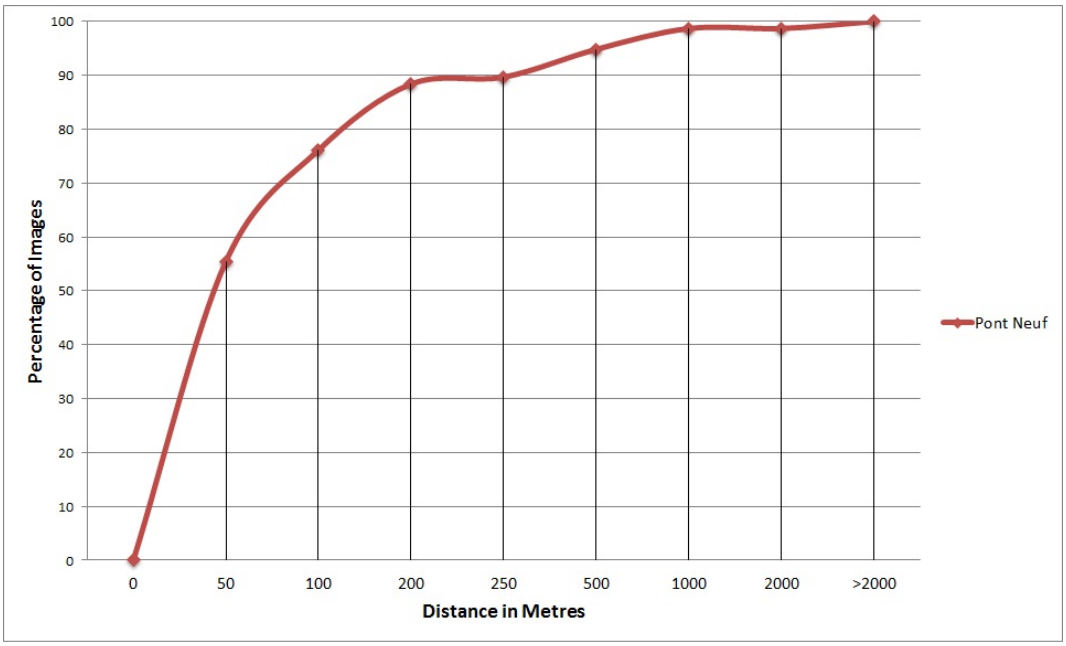

Figure 3.7: A graph illustrating the accuracy of the geo-tags for the images in the dataset taken of the Pont Neuf Bridge from the banks of the Seine. Over 85% of all images are accurate to within 200 metres.

lion Flickr images and organised them into Wordnet categories. They found that tags describing or representing specific locations (28%) were the most common types of tags followed by tags representing objects or artefacts (16%). Tags representing people (13%) were the next most popular category followed by tags representing events (9%) with tags representing time the least popular category (7%). Based on these statistics it would seem that Flickr users would first associate a place with an image when annotating it rather than an actual event that might be occuring within the image.

Kennedy et al. [Kennedy et al., 2007] analysed how textual tags could be utilised to extract place and event semantics within Flickr collections. They generated a representative cluster of images for each of 10 well known landmarks around the world using only similarities between textual tags as a baseline. Textual tags were then also combined with location and content information. They found that adding location information increased the precision by 30% and adding content information increased the precision by a further 45%.

3.3.3 Analysis of Community Contributed Textual Metadata

In this section, Flickr tags are analysed to ascertain their usefulness in aiding the classification of landmark images. While any additional contextual information regarding image clustering and classification will generally be useful to some extent, noisy contextual information can harm classification accuracy. It is important to measure the level of noise that exists within community contributed image annotations to gauge whether they might help or harm clustering and classification processes.

To analyse the relevance of the manually created textual tag annotations for use in this work, a small subset of images were chosen randomly from the Parisian dataset. To determine the semantic relevance of each tag, they have to be examined

manually which is a time consuming task. This subset consisted of 100 images, which should be enough to provide an estimation of the applicability of tags for retrieving visually similar imagery. The tags that accompany these images were extracted and analysed by the author. Each tag was given a relevance score between one and five, with one being deemed as a tag with the most semantic relevance to the landmark/location depicted within the image and five being the score given to a tag with the least relevance to the content of an image. The relevance scoring procedure is outlined as follows:

- **Relevance rating 1.** A score of one is given to a tag that contains a high level semantic meaning such as the name of the landmark or location. Vague location tags or tags that define a large area such as 'Europe' or 'France' are excluded. For a score of one to be given a tag, it must contain the name of the main landmark or small geographical area contained within the image such as 'Eiffel Tower' or 'Place de la Concorde'.

- **Relevance rating 2.** A score of two is given to a tag that contains a mid-level semantic description of the content within an image. If a tag describes the type of landmark or location depicted or describes some additional information describing the part of a landmark that is photographed, it receives a relevance score of 2. Some examples are: 'Cathedral', 'Facade' or 'Fountain'.

- **Relevance rating 3.** A score of three is given to a tag that contains low-level semantic information about an image or a describes a city scale location. A score of three is also given if the tag describes a landmark or location that is in the immediate vicinity (within 500 metre radius). Examples of a low-level semantic tag might be 'outdoor', 'sky', 'night', 'river' or 'park'. The tag 'Paris' would also be given a relevance score of three.

- **Relevance rating 4.** A relevance score of four is given to a tag with very little relevance to the content contained within an image. These tags might be vague geographical descriptions such as 'Europe', 'city' or 'continent' that provide little discrimination value. Other examples include tags that contain vague spacial prepositions such as 'view'.

- **Relevance rating 5.** A score of five is given to an incorrect tag or a tag with no relevance to the content of the image. These types of tags might be heterogeneous or possibly describe an event that is relevant to the annotator, but not useful from a larger information retrieval perspective. Common heterogeneous tags include 'holidays', 'honeymoon' and 'trip'. Some other common irrelevant tags include the brand names of the camera manufacturer used to take the photo such as 'Canon EOS' or 'Nikon D50'.

An example image displaying the relevance results from sample image is illustrated in Figure 3.8.

In total there were 918 tags associated with the 100 randomly selected images, amounting to an average of 9 tags per image. The relevance of these tags with the semantic content of each image was quite poor. Of these 918 tags, only slightly over 10% were given a relevance score of 1. The majority of tags were deemed as 'noise'. Over 40% of tags were given a relevance score of 5, which was deemed not to be useful for image similarity measurements. The results of this analysis is presented in Figure 3.9.

Another attribute that could be useful to match images in a large dataset, is the level of uniqueness that exists between tags. Ideally from the perspective of this work, a high degree of uniqueness in the textual tags in images that contain different content is desired along with a low degree of change between tags belonging to visually similar images. This measurement is roughly analogous to inverse document frequency in text based information retrieval, which is

Flickr Image:
2356646082_7cb91e68fa_b.jpg

Relevance 1	2	3	4	5
Tuileries	Garden	Paris	City	Trip
jardin des tuileries	Sculpture	Sky	France	Travel
		Art	Statue	Winter
		Louvre	Seine	Vacation
		Concorde	Architecture	History
			Captial	French
				Artist
				View
				Sightseeing
				Culture
				Tourist
				Abroad
				Street
				2008

Figure 3.8: An example of an image taken from Flickr with its provided textual tags along with a relevance ranking. A ranking of 1 being most relevant to the content of the image, with a ranking of 5 being least relevant. This image contains a sculpture that is located within the Tuileries Garden in the centre of Paris.

described in more detail in Chapter 7. The tag frequencies in the 100 images were analysed.

From the results of this analysis, illustrated in Figure 3.10, it can be seen that the majority of tags within the dataset are quite unique. Almost 300 tags had a frequency of one, and the number of tags dropped significantly with the increase of tag frequency, with the exception of two tags.

Two tags that did appear regularly were 'Paris' and 'France'. Paris appeared in 86% of images examined, while France appeared in 54%. In the absence of geographical data these tags could be very useful in the predicting an image's location. The appearance of 'France' would suggest the image's location down to a countrywide scale, while the appearance of 'Paris' would give an indication of

the location down to a citywide scale. More importantly, the appearance of both tags together would reinforce the assumption that an image was taken in Paris.

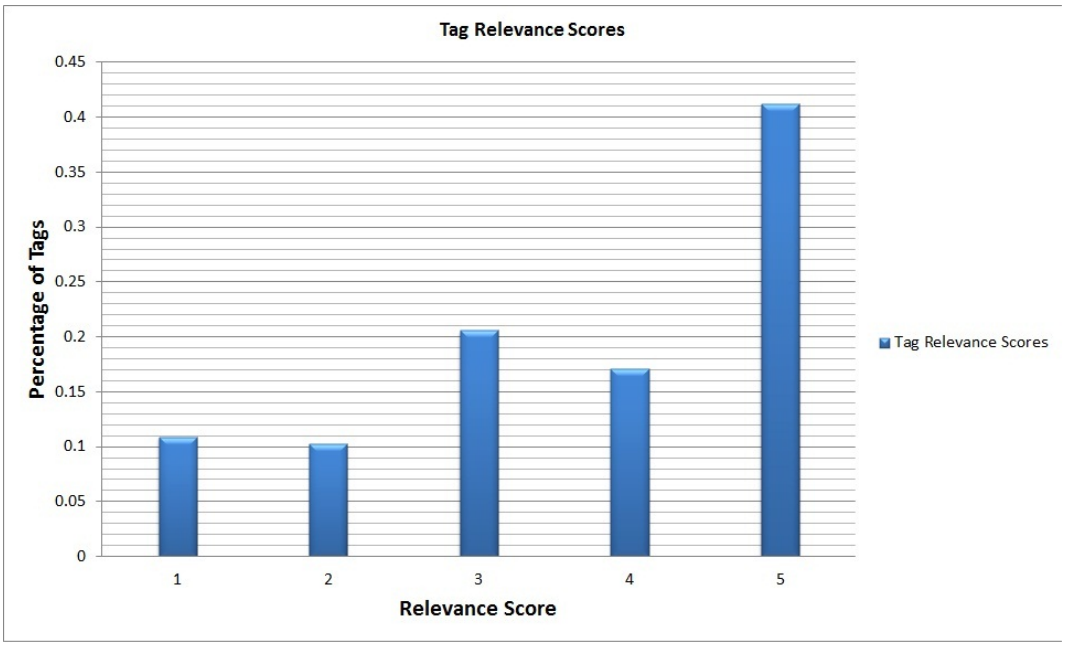

Figure 3.9: An outline of the results from the tag analysis experiments. The scores are based on the percentage of tags that received each relevance ranking score. As can be seen from this graph, the majority of tags examined were deemed to be noisy or heterogeneous, and not semantically relevant to the content of the image.

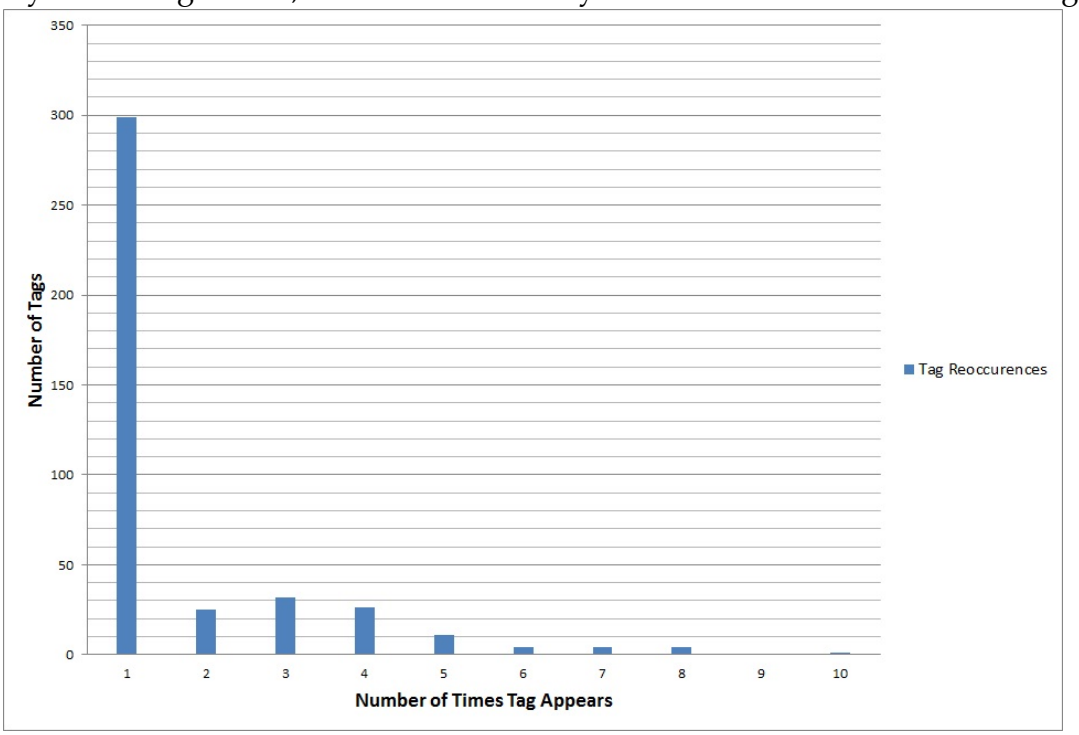

Figure 3.10: An outline of the analysis of the how unique user tags are. The scores are based on the number of tags that recur each number of times. As can be seen from this chart, the number of unique tags is quite high.

Chapter 4

Clustering Community Contributed Imagery

4.1 Introduction

In this chapter, several approaches to clustering large volumes of images are proposed and evaluated for the purposes of creating groups of visually near-identical images, which are to be used as inputs into machine learning algorithms. The chapter begins with an introduction to clustering algorithms. An outline of the proposed algorithm used in this work is then provided, followed by a description of all features used as part of this algorithm. The chapter concludes with an evaluation and the proposed algorithm is compared against an alternative widely used image clustering algorithm.

One important aspect of applying SVMs to solving the problem of landmark classification is having accurate sets of input data to train the models. Manually creating the training sets would be extremely time consuming, and in very large datasets infeasible. One approach to automatically creating accurate sets of input data is to cluster visually similar images, and use the results of this clustering process as the positive training data in the machine learning training phase.

The main aim of a clustering algorithm is to create groups of images that are similar internally, but each image in every cluster is clearly quite different from all images within other clusters. Clustering is a form of unsupervised learning, which means that there is no human supervisor assigning images to a cluster and that the clusters are organised in an automated manner, using some kind of similarity measure. Ideally this measure should produce clusters of images that replicate how a human supervisor might group sets of images into clusters.

Clustering algorithms tend to be quite computationally expensive, particularly when clustering imagery, as image features tend to be quite large in size. It is envisaged that one day the framework described in this work could be scaled up to contain a training collection of millions of images, therefore an efficient approach to clustering large amounts of community data is essential. There is no known clustering algorithm that is optimal for all uses and different algorithms perform better for different tasks. Due to the large scale of training data used in this framework (outlined in section 3.2.2), two attributes of any clustering algorithm must be taken into account: algorithm complexity and memory usage. Several clustering algorithms require that the number of final clusters be specified before the algorithm is executed. Without very expensive and time consuming human classification, there is no prior knowledge of the number of landmarks or clusters that will exist in the data, therefore many commonly used clustering algorithms are unsuitable for use here.

Clustering algorithms can be roughly grouped into two classes:

- **Flat clustering** Flat clustering is the creation of groups of clusters without a related structure. Each cluster is considered an independent entity and there is no information suggesting which clusters relate to each other. Several well known flat clustering algorithms include the k-means algorithm [Hartigan and Wong, 1979] and the Expectation Maximisation algorithm [Dempster et al., 1977]. Many flat clustering algorithms can be expensive,

and in many cases there is a requirement that the user has prior knowledge of the number of clusters that will be desired after the algorithm has finished completion.

- **Hierarchical clustering** Hierarchical clustering divides the dataset into a tree structure, which has many advantages with the datasets used in this work as it speeds up processing times. This hierarchical based clustering allows for the division of a dataset into much smaller clusters of data using efficient and inexpensive image comparison metrics. This division of data can continue iteratively down the tree structure allowing for more expensive and complex image comparisons to be processed at lower levels of the tree structure. Due to this iterative division, the most expensive operations can be carried out on the smallest number of images possible, similar to the divide and conquer paradigm used widely in computer science [Dwyer, 1987].

Flat clustering algorithms that linearly compare each image within a dataset to a potentially large number of cluster centres using expensive distance metrics (such as local feature point matching) are not scalable to very large datasets. A hierarchical clustering approach therefore, is adopted in this work.

4.2 Divisive Hierarchical Clustering

Hierarchical clustering algorithms can be separated into two main groups:

- **Agglomerative** Agglomerative algorithms follow a 'bottom up' approach to grouping data, where the dataset starts as single entities, with each datum representing a cluster in the initial stage. The algorithm then combines these clusters based on a merging criteria. This process iteratively continues from the bottom up until there is one single cluster. Commonly, four merging approaches are used;

1. *Single-link* is a method where the measure of similarity between two clusters is calculated based on the distance between their two closest features.
2. *Complete-Link* is a method where the measure of similarity between two clusters is calculated based on the distance between their two most distant features.
3. *Centroid* is a method where the measure of similarity between two clusters is calculated based on the average similarity between their features.
4. *Group Average* is a method where the measure of similarity between two clusters is calculated based on the average similarities between all features, including intra cluster features.

Agglomerative clustering algorithms typically have a quadratic running time $O(n^2)$, which generally makes them unsuitable for large scale clustering tasks, although recently several variants have been proposed with a sub quadratic running time [Walter et al., 2008].

- **Divisive Hierarchical** Due to the expensive running time of agglomerative clustering algorithms, an alternative to an agglomerative approach was adopted in this work called divisive hierarchical clustering (DHC), also known as a top-down clustering approach. The main concept behind DHC is that when the algorithm starts there is one large cluster of data, which is iteratively sub-clustered until a stopping criteria is met. This concept is illustrated in Figure 4.1. The stopping criteria that would apply to this framework would be the near-identical visual similarity of images within a cluster. This near-identical similarity is measured by comparing image regions using the SURF algorithm, and processing all images in the small-

est sub clusters in a graph linked by SURF correspondences. This SURF correspondence matching is described in more detail in section 4.3.9.

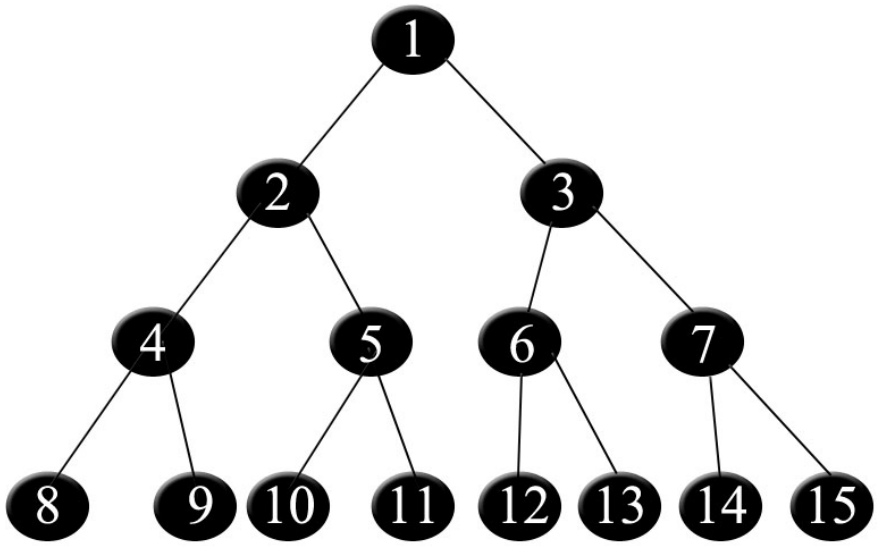

Figure 4.1: An sample of a divisive hierarchical clustering method. The dataset starts off as a single cluster which consists of all data (1), and is divided into sub clusters (2,3), based on some similarity measure, which in turn are sub-clustered (4,5,6,7) until a stopping criteria is met (8,9,10,11,12,13,14,15).

4.3 Hierarchical K-means Clustering

Hierarchical clustering approaches provide a solution to the shortcomings of the k-means algorithm by carrying out expensive matching processes on iteratively smaller datasets therefore reducing the numbers of expensive comparisons to be made. The problem still remains however, with how to efficiently cluster sets of data at each level of the hierarchical tree.

An hierarchical k-means (HKM) approach is one method to effectively cluster large amounts of data containing large variations in similarity. The HKM is an algorithm that consists of a tree structure with a branch factor of k, and a

number of levels l. To create the tree, data is iteratively clustered using the k-means algorithm. Each level consists of k cluster centres used as pivot points, which have been calculated using k-means on all of the data contained in its child nodes. Therefore, as one propagates down the tree, the size of the datasets used to calculate the cluster centres becomes smaller and smaller. The main idea behind the HKM algorithm is that at higher levels of the clustering tree where the data sizes are large, inexpensive distance measurements may be used to initially cluster the data. As one propagates down the tree, more expensive data comparisons may be carried out to generate accurate data groupings.

When using a hierarchical k-means clustering algorithm, two important considerations much be taken into account that determine the final number of clusters:

- The branch factor, or the cardinality of the clustering algorithm, referred to as k. This determines the number of clusters to be created in each iteration of the k-means algorithm.

- The number of levels in the tree, denoted as l. The number of levels in the tree will generally be directly related to the size of the data being clustered and the branch level assigned to the tree.

Without prior knowledge of the final number of clusters it is difficult to know in advance what values for k and l will be optimal for a given dataset in advance of a clustering process. In this work, different values for k were analysed and different values of l assigned depending on the value of k.

The k-means algorithm is an iterative algorithm that is non-deterministic, therefore there are no guarantees that optimal data convergence will be found. The iterative algorithm will continue until some criteria is met, which in this work is either when convergence is found or an iteration count is met. To speed up required processing time, a value of 25 is assigned to the maximum number of iterations.

Given a set of feature vectors $(v_1, v_2, .., v_n)$, the aim of the k-means algorithm is to partition this set of features into k sets of data to minimise the intra cluster variance and maximise the inter cluster variance. The algorithm comprises of 6 main steps:

1. Set an iteration count $iteration$ to 1
2. Choose a set of k random cluster seeds/centres $C_{iteration}$ $(c_1, c_2, .., c_k)$ from the global set of feature vectors
3. For each vector v_i, compute the Euclidean distance $dist(v_i, c_j)$, from each cluster centre $1, .., k$ and assign v_i to the cluster c_j where c_j is equal to $argmin(dist(v_i, c_k))$.
4. Increment $iteration$
5. Calculate a new set of means from each cluster and assign as new set of cluster centres $C_{iteration}$ $(c_1, c_2, .., c_k)$
6. Repeat steps 3, 4 and 5 until $C_{iteration} = C_{iteration-1}$ or $iterations >= 25$

Using this HKM algorithm, a wide array of distance measures using different features were evaluated in this chapter. Some of these distance measures included distances between low-level feature vectors and numbers of interest point correspondences. The distance measure used in the first stage of all HKM variants was based on geographical data.

4.3.1 Spatial-Based Clustering

By overlaying a grid over the surface of the earth, any location on the planet can be effectively described using a coordinate based system. The most commonly used geographical coordinate system in use today is the longitude/latitude system. Longitude measures distances between points on the earth's surface in a east-west

direction, while latitude measures points in a north-south direction. Combining two of these coordinates (one longitude and one latitude) can pinpoint any location on the planet accurately. These simple coordinates can be very effective from an image retrieval perspective to prune non-relevant images from a geo-tagged image corpus. These geographical coordinates are particularly effective for pruning datasets when the purpose behind the image retrieval involves searching for specific scenes and locations. It is for this reason that the first stage in the hierarchical based clustering approaches analysed in this work is based around spatial clustering.

Spatial-based clustering involves grouping numbers of images using distances between geographical data as the measurements to cluster the images. The dataset used in this work is quite large, and comparing large numbers of images is processor intensive and time consuming. Spatial data allows for filtering of unwanted images very efficiently, allowing the pruning of the candidate dataset, which in turn reduces the number of expensive image matching operations that need to be processed. Based on this knowledge, spatial based clustering is used as the first stage in the hierarchical clustering algorithm.

The first stage of a spatial clustering algorithm involves selecting initial cluster centres or seeds. Each of these seeds at this stage will simply be comprised of longitude/latitude coordinates in the WSG 84 format: dd.dddd [W3C, 2006]. To select these seeds a rectangle bounding box was created encompassing all of the images in the training set. This bounding box was created from analysing the minimum and maximum longitude and latitude coordinates from all images in the corpus. This bounding box was located at coordinates: This bounding box is roughly constituted with a width of 20km (18.80km) and a height of 11km (11.12km), which approximately consists of a total area a little over 200km^2.

The first level of all the hierarchical clustering approaches analysed in this work consists of clustering the whole dataset based on spatial data into k clusters

48.90 N, 2.27 E 48.90 N, 2.44 E

48.80 N, 2.27 E 48.80 N, 2.44 E

using the k-means clustering algorithm. The k-means algorithm as described in section 3.3 splits a dataset into k clusters based on distance measurements, in this case geographical distances, calculated using the Haversine formula (as described in section 3.3.1) from each cluster's centre.

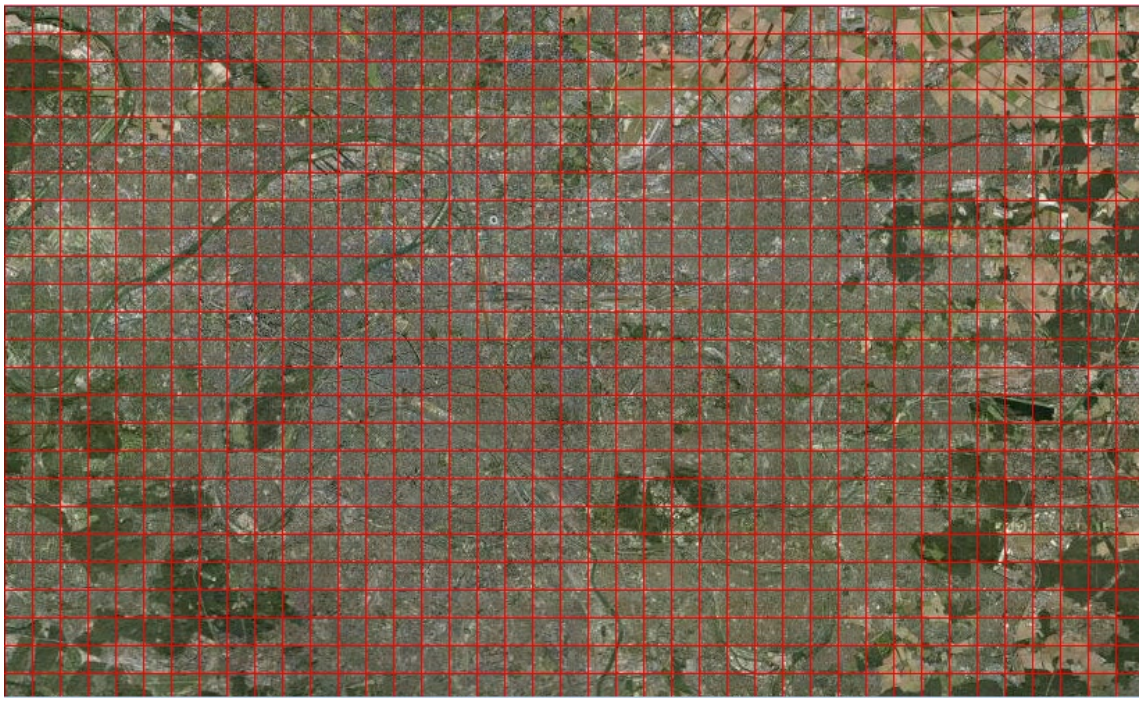

Figure 4.2: An illustration displaying the map area covered by the Paris dataset. The grid displays how the geographical space was partitioned into sub regions. The geographical centre of each of these segments were chosen as the initial cluster seeds for the k-means algorithm at the top level of the hierarchical clustering tree.

To provide initial cluster centres as inputs into the k-means algorithm, the geographical area was evenly partitioned in k subregions. An example of this partitioning is presented in Figure 4.2. The centre point of each region was chosen

as the initial cluster centre. All images in the dataset were then clustered using a k-means algorithm, and distances between geographical locations used to determine each images optimal cluster. Choosing an optimal value for k is a non-trivial task. It is necessary to analyse the geographical information to figure out how best to partition the dataset, to ascertain the effect that different values of k might have in pruning relevant and non-relevant images.

Analysing Effects of Spatial Radii

When using spatial data as a filtering mechanism to identify similar images, the aim is to choose a spatial distance that will contain the maximum number of correct matches, while filtering out incorrect matches. The number of images filtered out dramatically effects the speed and quality of a classification technique. Too many images retrieved will result in a low precision rate and a slow classification time. Too few images retrieved will reduce classification time but might result in a low recall rate.

To analyse the effects of selecting different spatial radii, 250 test images were selected at random from the corpus, and all images located within a number of different spatial radii from the test images were retrieved from the training corpus. For each of these searches, the average number of retrieved images was noted for each spatial radius examined. The results of these searches are presented in Table 4.1.

From section 3.3.1 it is known that on average over 80% of the images within this dataset have spatial data that is accurate to within 200 metres. This means that when selecting a spatial radius of 200 metres over 80% of the correct candidate images will be included after spatial clustering. Selecting a spatial radius of 500 metres should ensure that over 85% of the correct candidate images will be included after spatial clustering. Based on the results in Table 4.1, however, the number of images returned from a spatial query will have increased by over 200%.

Table 4.1: Analysis of Spatial Radii

Spatial Radius	Avg No of Images Returned	% of Training Set	Recall Rate
50m	1104	1.2%	55.2%
100m	1960	2.1%	75.1%
200m	3436	3.7%	80.9%
250m	4205	4.6%	82.0%
500m	8008	8.8%	85.8%
1km	16080	17.6%	89.0%
2km	29602	32.5%	92.8%

This means that the average processing time required to match a test image will have increased by over 200% even though the number of relevant images returned after a spatial query will only have increased by 6.25%. Clearly a threshold has to be selected which provides the best balance between precision, recall and processing speed.

By analysing the results displayed in Table 4.1 it becomes evident that a spatial radius of 200 or 250 metres would be the most preferable of all tested radii values, as these two spatial radii encapsulate over 80 % of the relevant images, while filtering out over 95% of all irrelevant images. Based on these experiments, along with the related analysis of geo-tag accuracy described in chapter 3 (section 3.3.1), a geographical radius of 250 metres is chosen as the optimal value for spatial based pruning in the remainder of this work.

Based on this information, the dataset was partitioned in sub-regions with a length of 500 metres $\times$ 500 metres, which meant that a value of 800 was chosen for k in the spatial clustering process. It must be noted that although 800 clusters might seem a lot for a first stage of a clustering algorithm comprising of a dataset of 90,100 images, many of these clusters contained very few or no images at all. On the other hand some of these clusters contain thousands of images in geographical regions where there is a high distribution of photographs.

4.3.2 Text-Based Clustering using Community Contributed Annotations

Introduction

User contributed tags have been used by several research groups over recent years as a means of clustering similar imagery. The most prominent example of this is the community tags assigned to images in the Flickr archive. Moëllic et al. [Moëllic et al., 2008] apply a Shared Nearest Neighbour algorithm to contextual data to cluster Flickr imagery. To measure tag similarities, they use the Pointwise Mutual Information measure which is defined as

$$pmi(w_i, w_j) = log(\frac{P(w_i, w_j)}{P(w_i).P(w_j)})$$

where $P(w_i, w_j)$ is the probability of tags w_i and w_j occurring in the same image, while $P(w_i)$ and $P(w_j)$ are the probabilities of w_i and w_j occurring in any image in the collection. These probabilities are calculated based on document frequency statistics. They combine this tag similarity metric with visual word histograms with a vocabulary size k equal to 5000, to cluster groups of images for three semantic queries: Eiffel Tower (location), Roger Federer (Personality) and Presidential (Event).

Abbasi et al. [Abbasi et al., 2009] measure tag and group frequencies to classify groups of Flickr images into landmark and non-landmark images. Images are classifed into landmark and non-landmark images using an SVM classifier combined with tag frequency statistics. They also propose a metric that they call the 'city tag frequency', which is a measure of the tag frequency within a specific city, which they use to classify a set of representative tags to describe a city.

Flickr tags are used by the Flickr retrieval system [Flickr, 2004], to return images that are semantically relevant to different queries. Queries that contain location information or some information relating to an event work quite well

using the tagging format. Tag similarity measures alone, however, contain no content information. While these tags are a good resource for extracting different kinds of information from image datasets, several issues arise in the context of image matching. In this work, the goal is to cluster imagery that is visually near identical (ie. Images that are taken of a landmark from a similar viewpoint, taken from similar distances and at similar zoom ranges) and these Flickr tags are created by human annotators and can be subjective. As can be seen from section 3.3.2, there is no guarantee that the tag provided by a human annotator is semantically relevant to the visual content of the image. Two images containing different near identical content could contain completely different tags, while two visually different images could have matching tags.

Other issues that can arise when using textual tags to measure visual or semantic similarity are polysemy and synonyms. Polysemy is when a single word can have multiple definitions, such as the word 'match' (This could indicate a sporting event, an object that closely resembles another or it could indicate a little piece of wood topped with sulphur that is used to ignite a fire). Synonyms are different words that have similar meanings for example: church, cathedral and chapel all indicate a place of worship for Christians. Users might describe an image using different synonyms. When comparing images based on similarity between tags, the context or semantic definition of these tags is not known in advance. Synonyms are problematic in a tag matching context as it is difficult when using text comparison methods to find a correspondence between two synonyms.

Tag Based Clustering

In this section, an evaluation is carried out using tag similarities as the distance measure as part of a HKM algorithm. A two-stage clustering method was processed. This consists of spatial clustering, described in section 4.3.1, followed

by clustering based on tag and image title similarities. These similarities are calculated by firstly processing all tags and titles into case insensitive tokens to account for similar tags being mismatched. Each token within a set associated with an image is then compared to each token contained within a set belonging to the cluster seed.

In order to compare the metadata, it is firstly organised into comparable groupings. The image title that is provided by a user can often describe the content of an image (Example: 'Me and Marie in front of the Eiffel Tower', would imply that the content of the image contains two people photographed with the Eiffel Tower in the background). For each image within the corpus, the image title is tokenised, and all tokens are added to the tag set associated with that image. All tags in this extended tag set are then grouped into a set S_t and this is then associated with the image.

There are many situations where tokens are slightly different but it would still be desirable for a match to be counted. For example, the tag 'EiffelTower' should be matched with the tags 'eiffel-tower' and 'Eiffel tower'. To account for these situations, it is necessary to process all tags within datasets a the view of normalising them to account for these small discrepancies between pairs of tags.

White space is removed within the sets of tags. Any tag containing white space is tokenised into separate tags, using the white space as the separator. Any tag containing an upper-case character in any position other than the first position of the tag was also tokenised, using the capital letter as the separating point. This was not the case if all characters within a tag were upper-case.

While there exist many grammatical rules to applying capital letters to words in language, there exists no enforced rules to ensure that users apply these to their metadata. Case-folding is therefore carried out on all characters within tags to lower case.

For each example image, the set S_t is compared against sets $S_{\{1.....n\}}$, where S_1 is equal to the set of tags belonging to the first image examined in the cluster, and n is equal to the number of images within the cluster. Several different distance measures to measure correlation between sets of tags were evaluated. These included:

- Jaccard Distance
- Dice's Coefficient Distance
- Overlap Coefficient Distance

Jaccard Distance

The Jaccard Coefficient is a measure of the similarity between sets of variables, while the Jaccard distance is a measure of dissimilarity [van Rijsbergen, 1979]. The formula for calculating Jaccard's Coefficient from two sets of tags, T_1 and T_2 can be described as:

$$c = \frac{|T_1 \bigcap T_2|}{|T_1 \bigcup T_2|}$$

whereas the formula for calculating the Jaccard distance from T_1 and T_2 can be described as:

$$d = 1 - c = 1 - \frac{|T_1 \bigcup T_2| - |T_1 \bigcap T_2|}{|T_1 \bigcup T_2|}$$

Dice's Coefficient

Dice's coefficient is a similarity measure similar to the Jaccard coefficient, however it does not penalise small numbers of similar objects to the same extent. It is defined as twice the shared number of items in two sets divided by the total number of objects in the two sets. It can be calculated from two sets of tags, T_1 and T_2, using the formula:

$$c = \frac{2|T_1 \bigcap T_2|}{|T_1 \bigcup T_2|}$$

In this work a dissimilarity metric is utilised to calculate the overlap distance with the following formula:

$$d = 1 - c = 1 - \frac{2|T_1 \bigcap T_2|}{|T_1 \bigcup T_2|}$$

Overlap Coefficient

The overlap coefficient is used to measure the overlap that exists between two sets. If one set is a subset of another, or the converse, the coefficient value is 1, whereas if there is no overlap between the two sets the returned coefficient value is 0. It can be calculated from two sets of tags T_1 and T_2 using the formula:

$$c = 1\frac{|T_1 \bigcap T_2|}{min(|T_1|, |T_2|)}$$

In this work a dissimilarity metric is utilised to calculate the overlap distance with the following formula:

$$d = 1 - c = 1\frac{|T_1 \bigcap T_2|}{min(|T_1|, |T_2|)}$$

4.3.3 Low-level Feature Based Clustering

Low-level image features are global features that consist of a single feature vector to represent an entire image. As each image is represented by a single vector, multiple images can be compared and matched quickly. One of the big disadvantages of global based features is their sensitivity to occlusion and variations in lighting conditions. Another significant issue with the use of low-level global features is that they do not discriminate sufficiently between inter class variations of images. Notwithstanding these issues, low-level features can be extracted and compared quickly due to their generally small vector lengths and relatively simple extraction methods. Combined with appropriate threshold values, low-level

image features can be successfully utilised to provide a method of eliminating non-relevant images at an early stage in a hierarchical clustering process. In this section, many low-level features and combinations of these features are explored together with varying threshold values to determine their usefulness for clustering near identical landmark imagery.

4.3.4 Colour Based Clustering

The first type of global features analysed for clustering purposes were colour image features. Colour features have been widely used in the past for comparing images for visual similarity [Ashley et al., 1995][Smith and fu Chang, 1996], and can be useful for filtering out unwanted images. The most basic colour feature is the colour histogram, which is a description of the distribution of colour features within an image. The colour space is usually quantised to provide more efficient memory footprints and faster feature matching. Each quantised colour or small range of colours are allocated to a bin within the histogram. The colour values of each pixel are analysed and binned according to their similarity to the colours associated with each bin.

One significant disadvantage with the use of colour histograms for image comparison is that feature vectors may not be discriminative enough to accurately match near identical imagery in large datasets. It is common that several visually different images can have similar colour histograms (see Figure 4.3 for an example). Another significant problem with low-level colour features in general is that they are generally not invariant to changes in illumination. Thus, images that are visually similar, and could be automatically classified as being similar by a human observer, may have very different colour histograms (for example, see Figure 4.4).

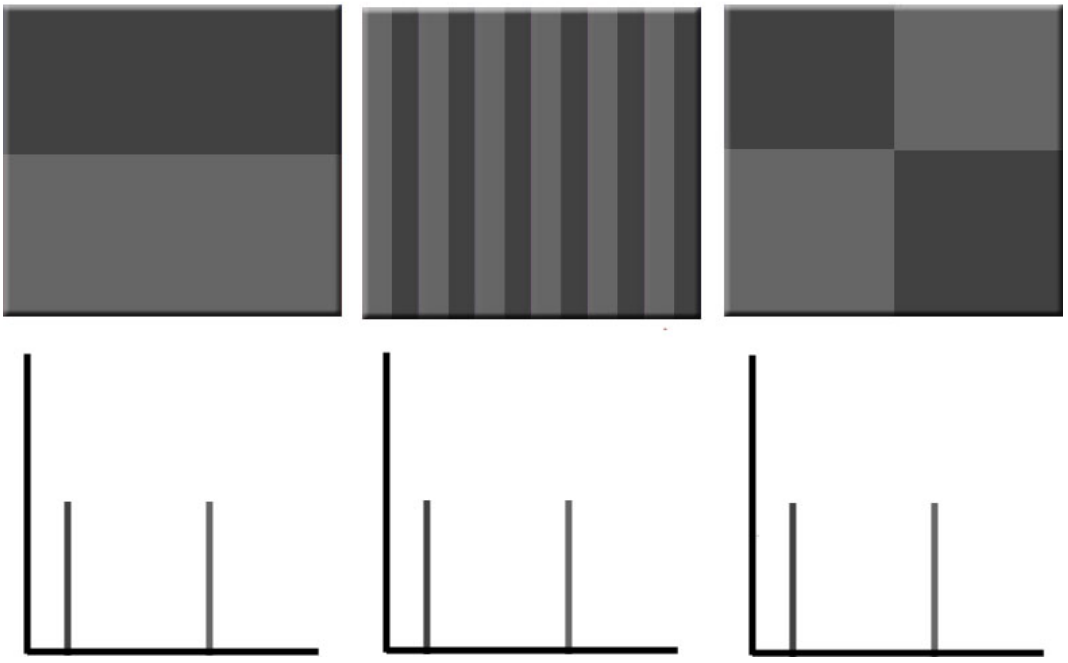

Figure 4.3: An illustration displaying 3 visually different images and their associated colour histograms. As can be seen from the illustration, all of the histograms are identical, even though the visual content of each image is different.

The main advantages of low-level colour features is that they require a small memory footprint, and quick to extract and compare. The problems with colour features arising from shortcomings in discrimination and invariance to illumination differences mean that a filtering threshold must be carefully selected to maximize the number of non-relevant images, while minimising the number of correct candidate images to be filtered out.

In this work, a colour feature based on the spatial relationships between similar colour regions is utilised. This feature is called a colour correlogram in literature.

Colour Correlogram

The colour correlogram is based on the computation of the spatial correlations between pairs of colours that exist within an image. The feature measures how these correlations change with spatial distance. This represents a significant im-

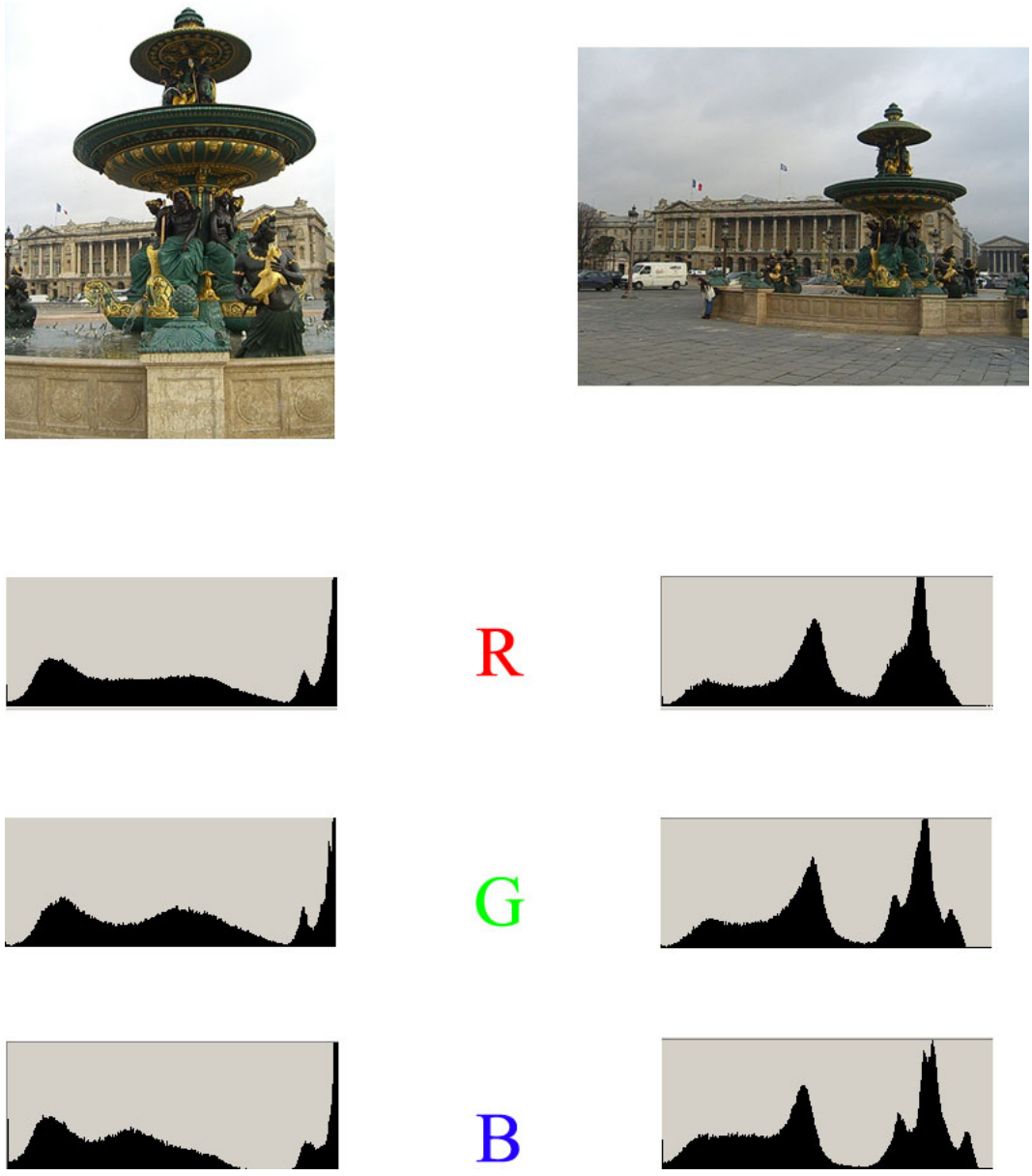

Figure 4.4: An illustration displaying the R,G and B histograms generated from two visually similar images with variation in illumination and a slight affine variation. As can be seen from the diagram, the three histograms for each image are radically different even though the image content is similar.

provement over a standard colour histogram feature and it is therefore evaluated in this chapter as part of a HKM clustering algorithm.

It has been shown that correlograms perform better in the HSV colour space than the RGB colour space for image retrieval tasks [Ojala et al., 2001], therefore in this work, the colour correlogram feature is calculated in the HSV colour space.

Firstly an image is converted from the RGB space to the HSV colour space using the following formula:

$R, G, B \in [0, C_{max}]$ where C_{max} is 255

C_{high} = max(R, G, B) , C_{low} = min(R, G, B) and $Chroma = C_{high}$

The saturation value is then calculated as

$$S = Chroma - C_{low} \text{ if } C_{high} > 0 \text{ otherwise } S = 0$$

The V value is then defined as

$$V = \frac{C_{high}}{C_{max}}$$

The H value is then defined as $H^{'} \times 60^{\circ}$ where

$$H^{'} = \frac{G - B}{C_{high} - C_{low}} \text{ if } R = C_{high}$$

$$H^{'} = \frac{2 + (B - R)}{C_{high} - C_{low}} \text{ if } G = C_{high}$$

$$H^{'} = \frac{4 + (R - G)}{C_{high} - C_{low}} \text{ if } B = C_{high}$$

The HSV image is then quantised into 256 colour values with 16 bins representing the H value, and 4 bins representing each the S and V values, in this case denoted as C_{1-256}. The correlogram feature is based on the probability that a pixel p_1 with a colour value c_i and another pixel p_2 with a colour value c_j are located at a distance d from each other. The standard colour histogram H can be defined for an image I as:

$$Hc_i(I) = Pr\lfloor p \in I_{c_i} \rfloor$$

whereas the colour correlogram can be defined as:

$$\gamma^{d}_{c_i c_j} = Pr\lfloor p_2 \in I_{c_j} || p_1 - P_2 | = d \rfloor \text{ where } P_1 \in I_{c_i} P_2 \in I$$

In this work a feature called an 'autocorrelogram' is utilised to speed up processing time. The autocorrelogram determines the probability that two pixels are identical at a distance d, ie. $c_i = c_j$ The feature utilised in this work used 4 values for d; 1,2,3 and 4 and produced a feature vector with a length of 256.

4.3.5 Texture Based Clustering

One of the most important classes of low-level features utilised for image comparisons and retrieval are those based around measurements of image textures. As noted in Chapter 2, several commercial image retrieval systems utilise texture features in their retrieval stages. This is mainly because many commonly used texture features are relatively quick to calculate and extract from images. They can be readily compared in a non processor intensive manner while still providing an accurate and useful visual description of an image or an image region.

A human observer can readily identify and recognise prominent texture patterns in an image (e.g. the stripes on a zebra or the pattern of leaves within foliage). It remains quite difficult however, to provide a concise definition as to what constitutes texture. Texture can be viewed as a measure of changes in light intensity within an image, the coarseness of these changes, or perhaps the repetition of intensity patterns across different regions of an image. In the absence of a standard definition, it is assumed that texture is some non random arrangement of intensity values. A number of approaches have been proposed and implemented to measure these arrangements. As described in Chapter 2, several of these techniques are widely used in the image retrieval and computer vision communities.

As image texture plays a fundamental part in many image matching and retrieval systems, in the following section, one of the most commonly used and discriminative texture descriptors is analysed to evaluate its attributes as part of a hierarchical clustering procedure.

Gabor Texture Features

Gabor texture features are a very commonly used class of texture feature in the computer vision community. They have been shown to outperform many

other types of texture features based on second-order statistics as described in Chapter 2. As Gabor filters have been utilised for a large variety of image analysis tasks such as fingerprint matching [Xu and Zhang, 2005] and object recognition [Jain et al., 1997], it is proposed in this work that they may also be useful as a discriminative mid level clustering feature. The Gabor features are evaluated as a cluster feature by themselves and as part of a fusion of different low-level features. The results of this evaluation can be seen in Tables 4.7, 4.11 and 4.12.

Gabor based texture features are a description of the coefficients obtained from a bank of Gabor filter responses in a range of different orientations, scales, and frequencies. This set of filters is calculated based upon the Gabor Wavelet Transform, which is formally defined as:

$$\gamma(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{\bar{x}^2}{\sigma_x^2}+\frac{\bar{y}^2}{\sigma_y^2}\right)+2\pi\sqrt{-1}Wx\right]$$

where

$$\bar{x}^2 = x\cos\theta + y\sin\theta$$

$$\bar{y}^2 = -x\sin\theta + y\sin\theta$$

σ_x and σ_y are scaling parameters that determine how large a neighbourhood around a pixel to calculate the summation for the filter response. θ is a parameter that specifies the orientation of the filter and W is the radial frequency of the sinusoid. By changing these parameters it is possible to create a bank of filters able to detect responses in a number of different scales, orientations, and frequencies. A texture feature composed of a suite of Gabor filters in 6 orientations, 3 scales, and 2 frequencies is used in this work producing a feature vector with a length of 36.

4.3.6 Clustering Based on MPEG7 Feature Sets

The MPEG7 standard is an attempt by the Moving Picture Experts Group (MPEG) to standardise a set of technologies that can be used to describe audio and video

content. These technologies consist of a set of tools for extracting multimedia features (visual and audio), along with standardised description schemes to describe these features. The well known MPEG1 and MPEG2 standards were mainly based around the representation of video data, however, the MPEG7 standard is more concerned with the description of the data along with its metadata. The MPEG7 standard aims to be applicable to a wide array of data types, such as audio, imagery and video. To achieve this, a number of different feature extractors were proposed depending on the underlying data type. These features can be broadly split into two main groups; audio based and visual based.

The visual features are split into four main types; colour, texture, shape, and motion. The motion features are intended to describe video files and require multiple images to be calculated, therefore they are not useful in this work and are disregarded. The shape descriptors are used to identify simple classes of objects, mainly based on the outline shape of an object. Landmark images such as buildings, for example, are likely to have similar outline shapes in many situations, and hence the discrimination value of these features for the purposes of this work is limited.

Colour and texture features are more likely to be useful for the purposes of this work and therefore it is these features that are analysed and evaluated.

Scalable Colour Descriptor

The scalable colour descriptor (SCD) is calculated based on the Haar transform of a colour histogram in the HSV colour space. The SCD feature has been widely used in image matching and image retrieval systems in the past [Chatzichristofis et al., 2009]. To calculate the SCD, firstly an image is mapped to the HSV colour space using the formula described in section 4.2.4. A Histogram with 256 bins is then extracted from this image, with the H component quantised to 16 bins, and the S and V components quantised to 4 bins each. This histogram

is then normalised and mapped into a four-bit integer, with more significance given to smaller values. This allows for efficient storage requirements. The Haar transform is then applied to each of these four bit integers across all values in the H, S and V bins.

It is possible to sum the values of every two adjacent Hue features to produce a quantised histogram of length 128 with 8 bins given to the H component and 4 bins given to the S and V components respectively. If desired, this process may be repeated allowing for the creation of further quantised histograms of lengths 64, 32 or 16. While these quantisation levels allow for faster matching of histograms and smaller memory requirements, the smaller the histogram length, the lower discrimination value of the feature. Therefore, in this work, a histogram length of 256 is used to ensure the highest level of discrimination.

Edge Histogram Descriptor

One of the most commonly used texture features in the MPEG7 standard is the Edge Histogram Descriptor (EHD). The EHD is a global based feature vector containing spatially organised histograms of edge orientations detected within an image. It is based on the measurements of four directional edges (vertical, horizontal, 45° and 135°) and one non directional edge.

When extracting image features to represent objects or landmarks within an image, it is preferable to have some division of the image into meaningful regions that are relevant to the actual objects/landmarks depicted. Once this division has been calculated, features can be calculated for each region, which allows for the inclusion of local information to be embedded into these features.

Global based features traditionally were calculated based on the whole content of an image. The main disadvantage of this is that all geometrical information regarding the layout of the extracted features is disregarded. It is preferable for global features to include some spatial information, which also increases the

discrimination value of the feature. Two of the most commonly used approaches to dividing an image into these regions are: Segmentation based and Block based division.

- **Segmentation based** division involves utilising a segmentation algorithm to partition the image into non-uniform segments that are relevant and display shape similarity to the object within the image. Optimally segmented images would provide a lot of additional geometrical information about objects depicted, such as shape, size and position. Object segmentation is still a active research field however, and there is no one algorithm that will lead to optimal segmentation in all situations [Gokalp and Aksoy, 2007]. Some alternative approach is therefore desired. One alternative of many is block based segmentation.

- **Block based segmentation** is the process of partitioning the image into blocks or regions, each one a predetermined size, and calculated in a defined manner. Each of these blocks is then treated as a separate entity for the purposes of feature extraction and the geometrical information regarding the regions location and relationship to other regions can be preserved in the feature descriptor. This information provides a weak form of geometric consistency when comparing and matching features from multiple images. The MPEG7 edge histogram feature utilises a block based segmentation scheme.

To calculate the feature, the image is firstly partitioned into 4×4 (16) equal sized sub images. The width and height of each block is $W/4$ and $H/4$ respectively, where W and H represent the overall width and height of the image. Each of these sub images is then treated as a separate entity. Irrespective of the size of the image, each of these blocks are further divided into 1100 smaller blocks (experiments carried out showed that a value of 1100 small sub blocks seemed

to capture good directional edges [Manjunath et al., 2002]). Each of these smaller sub blocks are then processed with a suite of 5 oriented edge detectors (0, 45, 90, 145 and non-directional). The sub block is then marked as the orientation that had the maximum edge strength outputted from these edge detectors, if above a threshold. If not, the block is disregarded. For each original larger sub block (16 in total) the average numbers of edges in each orientation is histogramed into 5 features. As this process is repeated for each larger sub block, this gives a total of 80 values to create the global EHD. This process is illustrated in Figure 4.5.

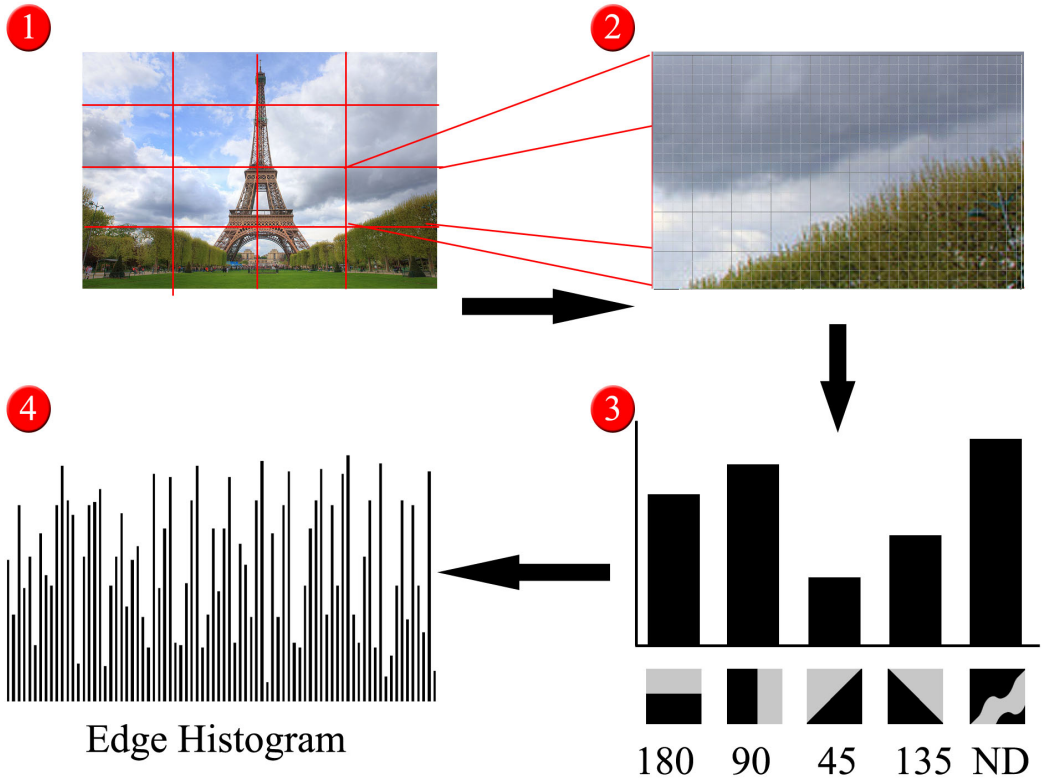

Figure 4.5: An illustration displaying the process of extracting an edge histogram feature from an image. Firstly the image is split into 16 sub images (1), followed by the further block segmentation of each of these sub blocks to 1100 much smaller blocks (2). A histogram is created for each large sub block, containing 5 values (3). All of these smaller histograms are merged into one global histogram (4).

4.3.7 Hybrid Low-level Feature Based Clustering

While a single global image feature may not be discriminate enough for large scale accurate clustering, it is possible that combinations of global features might perform a lot better. It is therefore necessary, to evaluate different combinations of these features to ascertain which combination might perform best for landmark viewpoint clustering. Four combinations of global features were evaluated;

- Spatial Colour and Texture
- Colour and Texture
- Colour and Edge
- Texture and Edge

All of these hybrid approaches were located at level 2 and 3 of the HKM algorithm. The first level of the hierarchical clustering algorithm for each evaluated hybrid approach consisted of k-means clustering using spatial data with a value of 800 for k. Additionally for each of the low-level hybrid approaches examined, the feature vectors for both features were normalised.

4.3.8 Inverted Visual Word Features

As shown in the past, low-level image features are not as discriminative as local image features [Bosch et al., 2006]. It is assumed that a more successful clustering algorithm could be created based on these local features. Local image feature matching using brute force techniques is slow to process and with large clustering tasks the processing time required on the full datasets would be undesirable.

A technique has been proposed in the computer vision field that allows for the quantisation of these local image features into a single global feature, while still retaining some of their discrimination value. This level of discrimination can be

roughly correlated with the level of quantization. As the quantization increases, the discrimination values decrease. This is not always a negative attribute (for example, in the case of scene classification, a higher level of quantization adds robustness to classification models). However, for the purposes of clustering, a happy medium is desired that can be used as an efficient method to cluster images at a high level of the hierarchical pipeline. The level of quantisation was carefully chosen as to maximise the number of similar images within a cluster, retaining a high level of recall while also disregarding the maximum number of dissimilar images. In this work a vocabulary size of 50,000 was used.

Images are clustered with the cluster centre that they share the most inverted visual word features with. Soft assignment is used in this method, which means that an image may be clustered with more than one cluster centre provided that the number of correspondences is above a threshold of 20. Once this clustering has taken place, images within each cluster are ranked in descending order by the number of visual word correspondences. The top k nearest neighbours were then retained for use in a final local image feature verification stage. All images ranked beneath k were disregarded from the cluster.

4.3.9 Local Image Feature Clustering

The lowest level of the hierarchical tree structure consists of expensive comparisons of local image patch features (SURF). As local image features generally outperform global images features for image matches purposes, this stage is the most crucial to obtain a high level of precision in each cluster.

Point to Point Image Comparison

As described in Chapter 2, the SURF algorithm (and many image patch description algorithms in general) will output a highly discriminative feature vector,

describing intensity changes in the local region surrounding the actual detected point. An accurate method to compare features is therefore desired.

As the SURF algorithm outputs a 64 value feature vector for each point, it would seem logical to utilise a standard distance measurement when comparing two features such the L1 Norm:

$$D(v_1, v_2) = \sum_{i}^{n} |v_{1_{(i)}} - v_{2_{(i)}}|$$

or the L2 Norm:

$$D(v_1, v_2) = \sqrt{\sum_{i}^{n} (v_{1_{(i)}} - v_{2_{(i)}})^2}$$

where v_1 and v_2 represent two interest point descriptors and n represents the length of the feature, which in the case of SURF would be 64. It has been demonstrated by Lowe [Lowe, 2004], however, that these standard distance measurements are particularly sensitive to small changes, such as affine transformations. Due to these sensitivities, Lowe suggested utilising a method called the distance ratio test to compare and match two local image features.

The distance ratio test is a method based upon the ratio of distances between the two nearest neighbours to a point. If the ratio of the Euclidean distances between features v_1 and v_2 (where v_1 is the nearest neighbour to a test point and v_2 is the second nearest neighbour) is above a threshold, the test feature and point v_1 are considered a match. Lowe found that using a threshold value of .8 eliminated 90% of the false positive matches. He also suggest that for 2 images to be considered a match or at least an object depicted within two images, that it required 3 of these distance ratio matches to be considered a good match.

Geometric Consistency

SURF image features are quite discriminative and invariant to a certain extent to occlusion, affine,rotation and scale variations [Bay et al., 2006]. It has also been

shown that matching approaches using the distance ratio test are invariant to a certain level of image noise [Lowe, 2004]. However, even with all these properties, it is still quite possible that in very large collections of images there will be false positive matches between sets of SURF features. One approach to reduce the percentage of these false positives, is to determine or verify that sets of SURF correspondences share the same geometric properties. An object should retain geometrical properties irrespective of the scale, viewpoint or orientation that it has been photographed from. The same remains true for sets of local image features extracted from a landmark within an image. The sets of matched features between two landmark images should share geometrical properties, even if they were photographed at different scales and from different viewpoints. For example, a landmark containing straight line features, will still contain straight line features when photographed from a different scale or rotation.

Using sets of SURF correspondences, it is possible to estimate a geometrical relationship between a landmark depicted in two or more images based on the geometrical properties of the matched features. By ensuring that all matched features adhere to this relationship, it provides a higher level of discrimination by verifying that the matched keypoints detected from one image correspond geometrically to matched keypoints extracted from the same landmark within another image. This process is referred to as geometric consistency or geometrical verification, and it has proved successful in verifying that two matched images do indeed contain the same matched objects [Fan et al., 2006].

Random Sample Consensus Algorithm

One approach to calculating geometrical models between sets of image features and the approach adopted in this work, is the Random Sample Consensus (RANSAC) algorithm [Fischler and Bolles, 1987]. The RANSAC algorithm is an iterative algorithm, that can estimate a geometrical model from a set of point

correspondences between two images. It is assumed that any matched pair of images will contain outlier matches, possibly due to occluding objects, repetition of a structure pattern, or false positive matches between SURF features. One big advantage of the RANSAC algorithm is the ability to remove these outlier matches that do not correspond to the geometrical model between a pair of images.

The RANSAC algorithm takes as input a set of features N, where each feature is a correspondence between two matched SURF features. It is assumed that the algorithm can calculate a model based on a set of features S, where S is a subset of N. In this work, the value assigned to the size of S is four, which is required to calculate a homography between a pair of images. To estimate the geometrical model, the algorithm comprises of five main steps:

1. Calculate a random subset of features S from N
2. Estimate a model based on the set of data S
3. Calculate the number of features from the total dataset N that fit the model and call this number m.
4. If m is above a predetermined threshold parameter, fit the model and exit algorithm, else if $m >$ previous value of m, mark this model as the best fit and continue algorithm.
5. Repeat steps 1 to 4 k times, where k is a predetermined parameter
6. Return the best fit model

RANSAC is a non-deterministic algorithm, and therefore some sort of stopping criteria is required. The parameter k determines the maximum number of iterations that the algorithm should repeat itself. In this work, an approximate estimate is made that 50% of all SURF matches are inliers (ie. actual matches between landmark objects), therefore a value of .5 is assigned to a parameter o. As

the algorithm is non-deterministic, it will produce a reasonable result only with a certain probability, defined as p. Although usually an acceptable probability value of 99% is the norm, the number of iterations required to estimate a model with a probability of 99% is large (72 iterations in this case) and thus the time required to verify two images (which will be repeated a large number of times) is not acceptable. In this work, it is deemed that a desired probability that a correct model be found of above 90% to be an acceptable value. This value was chosen to reduce the size of k and therefore reduce the processing time involved, while still retaining a reasonable probability estimate. To calculate the number of iterations, the following equation is solved for k:

$$1 - (1 - o^m)^k >= p$$

where m is the size of the subset N (ie. the number of inliers required to estimate the model). In the context of this work, this equation then becomes

$$1 - (1 - .5^4)^k >= .90$$

which is turn gives a value of 36 for k, which exactly halves the number of iterations if the desired value for p was 99%

$$1 - (1 - .5^4)^{36} = .903$$

Once a model has been fitted, all data points are verified against the model and outliers are removed, where outliers are determined to be data points that don't fit the model. All inlier features (ie. data points that fit the model) are deemed to be geometrically verified.

It must be noted that one big disadvantage of the RANSAC algorithm is that it can require a lot of processing time. When this process has to be repeated a large number of times, it can provide a processing bottleneck that will lead to unattractive matching and clustering times. When comparing very large numbers of images, this could be intractable using today's hardware. It is imperative,

therefore, from a image matching and clustering perspective that geometrical verification is carried out only on a small subset of candidate images, as opposed to large portions of the training corpus.

Pruning Cluster Outliers

Once there is a small number of images located in a cluster at the lowest level of the clustering tree structure, it is necessary to carry out a verification procedure that might remove irrelevant images from the cluster. From experiments carried out as part of this work, it was determined that SVMs provide a higher level of classification accuracy, when the training data has a small spectrum of affine variation. If the training images contain a wide affine variation there can be a significant decrease in classification accuracy. By carrying out a verification stage, the aim is to remove any outlier images that might have a large affine variation from the other images within the clusters. To carry out this verification, a graph data structure based on geometrically verified SURF correspondences was used.

Graph Data Structures for Evaluating Cluster Correspondences

A graph data structure is a commonly used data structure in computer science that consists of a collection of linked data in a non structured manner. A graph is quite similar to a tree in that nodes are linked to one another, however a tree contains nodes that are linked in a hierarchical manner where each node except the root has a parent node. Graphs are different in that there does not necessarily have to be an hierarchical structure and graphs can generally be bi-directional or multi-directional.

The aim of this section is to describe processing at the very lowest level of the clustering procedure that is utilised in this work. At the lowest level of the hierarchical tree, it is expected that there remains a large number of small clusters containing subsets of the overall dataset. Due to the small size of these subsets,

it is possible to carry out expensive processing tasks to verify that an image optimally belongs in the cluster to which it has been assigned.

To carry out this verification, it is not sufficient to simply compare all images against the pre-determined cluster centre. The centre of each cluster was assigned at a previous level of the hierarchical algorithm (such as that assigned using inexpensive global image features) in a different feature space. This might not represent the optimal representative cluster centre. One option at this point is to randomly select a cluster centre and measure correspondences between this center, or to perhaps select a representative image (perhaps based on the image with the highest number of feature matches between all images with the cluster) which has a high probability of being the most iconic image in the cluster and then compare against this representative image. There are disadvantages however, to both of these approaches.

Due to experiments carried out in this work, it has been established that classification models perform more accurately when the collection of images used in the training phase for each class contains a small spectrum of affine variation. Based on this, the overall aim of the clustering pipeline is to cluster images that have been taken from a similar viewpoint (desired to be in an affine variation range of approximately 45 degrees, and in a similar scale). The main disadvantage of selecting representative images or selecting a random cluster centre at this stage is that, although the iconic image could be representative of the entire cluster, it is still likely that images within the cluster might be matched with the representative image, but might be taken from a viewpoint outside of the desired range.

A method is required that will not only compare an image within a cluster to the cluster center, or the 'iconic' image within the cluster, but to compare each image to all images within the cluster. This is to establish whether an image is an outlier within the cluster, that might simply be matched to the cluster center. For the purposes of this section, an outlier is defined as an image that is visually

dissimilar, or has been photographed from a different viewpoint than the majority of images within the cluster.

The approach adopted here is to organise all images within a cluster into a graph structure G_k, where k is equal to the number of images within the cluster. Each image represents a node N within the graph, and there is an edge N_{ij} between nodes N_i and N_j if there are at least 3 geometrically verified SURF matches between the images. Each node is then examined and any node containing a number of edges below a threshold t is deemed as an outlier and removed from the cluster. From empirical evaluations, a value for t of 3 appears to perform well, and was chosen as a good balance between retaining candidate images within a cluster and removing obvious outliers. This process is illustrated in Figure 4.6.

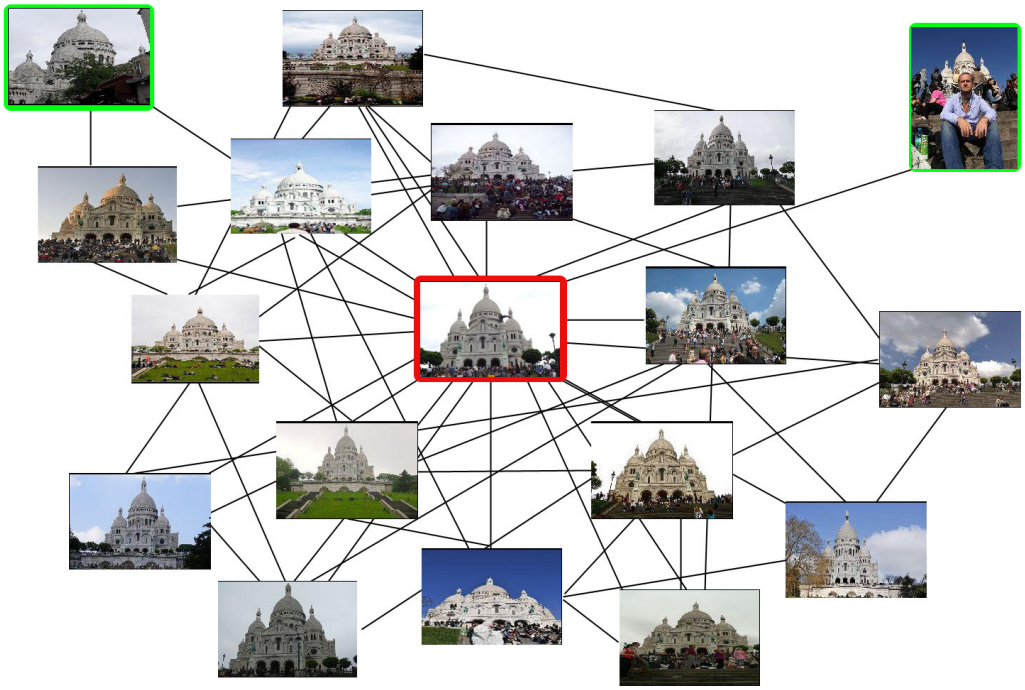

Figure 4.6: An illustration displaying the process of pruning outlying images from a cluster at the final level of the clustering process. This illustration contains a subset of the cluster created in the evaluation process using the red bordered image in the centre as the cluster seed. The two green bordered images were eliminated from the cluster based on the number of edges that their nodes contained within the graph. In total, this cluster contained 36 images after outlier pruning.

4.4 Clustering Based on Hashing Techniques

In this section, a widely used image clustering algorithm is described, implemented and compared against the hierarchical approach to evaluate performance. This approach has been proposed in the literature for efficient clustering of large scale image collections, specifically focusing on near-duplicate image clustering, using hash functions [Foo et al., 2007][Chum and Matas, 2008][Frahm et al., 2010]. Specifically, hash functions that allow for fast approximations of nearest neighbours in feature space. This technique is called Locality Sensitive Hashing (LSH). In this section, an implementation of a LSH algorithm was implemented and the hierarchical clustering approaches described in section 4.3 are evaluated against this implementation.

4.4.1 Hash Tables

Large scale image clustering is an computationally expensive process, due to the time required to compare large numbers of image feature vectors, therefore a technique that accurately approximates visually similar images from a large corpus in near constant time would reduce the processing time significantly (from $O(knm)$ in the case of k-means to $O(1)$).

A hash table is a data structure that enables rapid mapping between a key (eg. a string) and associated value. This achieved by using a 'hash function', that maps the key to an integer that is used to index a table storing the associated values.

Linear search allows for searching for an object in O(n) time where n is the number of objects in the database. For a large scale database, this could be infeasible for many applications, particularly, real-time image matching applications. A balanced binary search tree structure allows for search in O(log n) time as the search space is halved at each level of the tree. A well designed hash table, however, will allow for the lookup of a key value in O(1) time. The speed gain

associated with the use of hash tables over other data structures therefore is significant.

A hashtable consists of an array of values with each element in the array (also called a bucket) storing a value, or a group of values. A key value is associated with each object to be inserted into the hashtable. Each key value is mapped to a location within the array. This mapping is created with the use of a function called a hash function. The output of this hash function is called a hash value and will be in the range 0 - N where N is the size of the hash table size. Ideally a hash function should provide a uniform distribution of hash values and ensure that two distinct key values get mapped to different buckets in the hashtable (with a high level of probability).

If a hash function returns the same hash value for different key values, a collision occurs. Even with a good hash function and a uniform distribution, collisions are inevitable. The load factor of a hashtable is the ratio between the number of stored items and the size of the table. A low load factor can lead to less collisions, however there are significant memory overheads. Most hashtable structures are dynamically resized based on the load factor of the structure. Once a load factor threshold is passed, resizing is carried out to prevent high numbers of collisions.

4.4.2 Locality Sensitive Hashing

Locality sensitive hashing (LSH) is an algorithm based on hash functions that allow for approximate nearest neighbour searching in sub linear time. In this section, a clustering approach based on LSH is implemented and evaluated. The main idea behind LSH is that when a feature is close to another in feature space, after a projection, the two features will remain close together.

LSH provides the ability to very quickly cluster an image based on the values of initial cluster seeds (in O(n^{1+p} log n)), as opposed to iteratively comparing a feature against all cluster seeds such as in traditional k-means clustering.

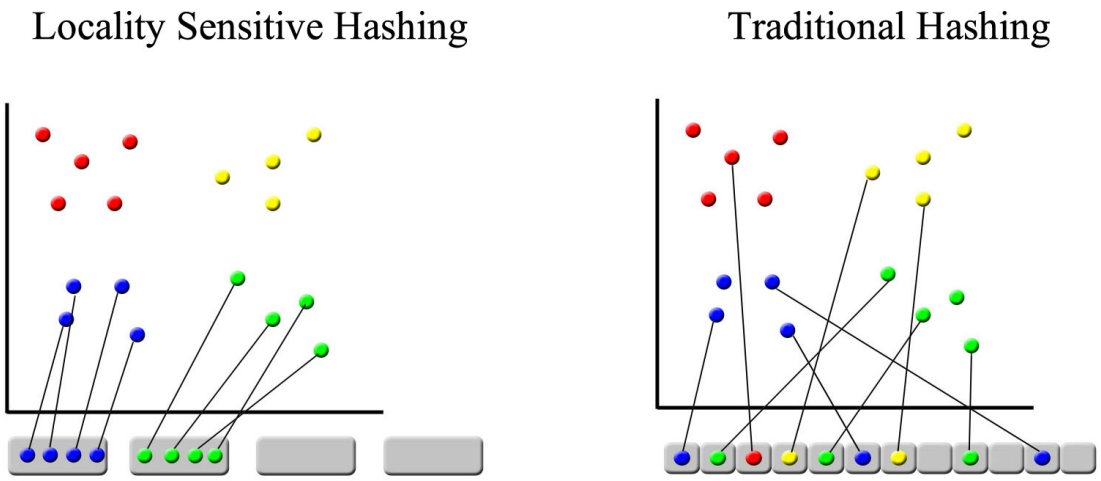

Figure 4.7: An diagram illustrating the main concept behind Locality Sensitive Hashing. As opposed to standard hashing techniques, features that are close to each other in some feature space will be assigned to the same hash bucket.

4.4.3 Hashing Functions

The main idea behind locality sensitive hashing is that a feature v_1 will be projected to another feature space using some hash function. Another feature that is located close in feature space v_2 will also be located close to v_1 after this projection. Once these projections have taken place it assumed that both features v_1 and v_2 would be hashed to the same bucket, which is illustrated in Figure 4.7. In this work, a hash function was implemented that was based on p-stable distributions described in [Datar and Indyk, 2004].

4.4.4 BOW Feature Histograms

Similarly to LSH implementations described in [Lee et al., 2010], each image was represented as a set of Visual Bag of Words (VBOW) features, which have been

used widely in image retrieval tasks in recent years. They provide a global based alternative to large numbers of local image features , and have been shown to be robust and discriminative [Sivic and Zisserman, 2003]. The VBOW histogram is a quantisation of a finite number of local image features into a single global histogram. In the VBOW model, a static collection of interest point descriptors is used as a comparison set called a vocabulary. A histogram is created with a length k, where k is the total number of features in the vocabulary. Interest points from test images are then compared against this vocabulary and the histogram bin associated with the nearest neighbour in the vocabulary is incremented.

4.4.5 LSH Parameter Selection

When using VBOW, the main parameter to optimise is a value for k, which determines the length of the vocabulary to use, and in turn, the length of the histogram vector. Another parameter to select, is the number of hash tables to utilise called C. A higher number of hash tables should increase recall, however they will effect precision values negatively. Additionally the higher the value for C the higher the memory requirement. To account for memory restrictions, some techniques have recently been evaluated based on large scale parallelisation of hundreds of desktop machines [Aly et al., 2009]. In this work, four values were evaluated for C; 16, 25, 45 and 60.

4.5 Clustering Evaluation

In this section, an evaluation is carried out on several different hierarchical clustering approaches. It is important that a clustering algorithm performs quickly, however, as the clustering processes are usually carried out offline, a more important measure is the accuracy of the final result. It is not useful to cluster images using a very fast performing algorithm if the outputted results are full

of inaccurate groupings. If the algorithm does not perform accurately, the input classes used to train classification models in Chapters 5 and 6, will be very noisy. Depending on the levels of inaccuracies, results would be expected to be no better than random. It is important, therefore, to compare and evaluate many different clustering approaches to ascertain which one will provide the highest level of intra cluster similarity and the lowest level of intra cluster similarity with the datasets used in this work.

It is quite challenging to evaluate the outputs of clustering algorithms. It would be very time consuming to manually inspect thousands of clusters outputted from a number of algorithms. Alternative approaches to ascertaining clustering accuracy, such as comparing community provided contextual information as a verification of accurate clustering, is fraught with potential problems. There is no guarantee that the labelled data is accurate, and hence will lead to inaccurate evaluation results when using metrics such as precision or recall. Another challenge in evaluating clustering approaches in this work is that all of the hierarchical approaches use the k-means algorithm. This is a non-deterministic algorithm that might produce different cluster results on each run, as the cluster seeds are chosen randomly before each initialisation of the algorithm. Due to this observation, it would not be an even comparison evaluating two different clustering runs using different features or distance metrics.

An information retrieval evaluation methodology is adopted in this section. A benchmark of an optimal clustering process was processed on the dataset. This benchmark process used pre-determined cluster seeds at each level of the hierarchical tree, which allows for a like for like comparison between different clustering approaches. Once this benchmark has been created, all evaluated approaches are evaluated against the benchmark, using a wide variety of common information retrieval metrics.

To create this benchmark clustering set, a brute force processor intensive approach was used which, should be optimal for clustering images in large datasets. This approach should be optimal, as in the majority of the best performing clustering runs, this brute force matching approach is either approximated or images are pruned out before this expensive brute force matching is carried out on subsets of the corpus. It therefore stands to reason that a brute force approach on the full dataset should outperform brute force searches on possibly inaccurately pruned subsets of the same data. Due to the processing time constraints, this benchmark was restricted to 250 clusters from the Paris dataset.

4.5.1 Benchmark Clustering

To ensure that each clustering run produced comparable results, 250 random images from the dataset were randomly chosen as initial cluster seeds. These same 250 images are then chosen as cluster centres at each stage of the clustering processes. For each of these seed images, all images in the corpus within a spatial radius of 250 metres were examined and clustered. One of the most accurate methods to match and cluster similar images is to use a technique called point to point matching using local image features followed by a geometric consistency check. The idea behind point to point matching is that each interest point feature extracted from an image is compared against every interest point extracted from another image. Point to point matching using extracted SURF interest point features was is carried out. This matching process used the distance ratio test with a ratio of .8, similar to the ratio used by Lowe [Lowe, 2004] with SIFT features for image matching. All images were clustered based on the number of point to point matches between the image and the cluster seed image. If there was no point to point match between an image and all of the cluster seeds, the image was disregarded.

This point to point matching phase was then followed by a geometric verification stage to verify that all clustered matches are geometrically consistent. This stage was carried out after initial clustering due to processing time constraints. The algorithm used in the geometric verification stage is the Random Sample Consensus (RANSAC) algorithm. For each cluster the images were ranked based on the number of geometrically consistent matches between each image and the cluster seed, if the number of geometrically verified matches were above a threshold of 3, the image was clustered with the seed image, otherwise, it was disregarded. As this benchmark clustering method is near optimal with regards to current state of the art image comparison methods, all evaluated approaches were compared against it.

4.5.2 Evaluation

The evaluation is based upon several commonly used information retrieval evaluation metrics; precision, recall, and the F-measure. In this clustering evaluation, a relevant image is defined as a clustered image that is also contained within the associated optimal cluster. If an image has been grouped with a cluster centre using the optimal point to point matching technique, and then also grouped with the same cluster centre using an hierarchical approach, it is deemed relevant.

Precision is a commonly used evaluation metric that measures how many members of a retrieved set are relevant. Therefore, precision is defined as the fraction of relevant clustered images, divided by all images within a cluster. It can be formally defined as

$$P = \frac{Number\ Of\ Relevant\ Images\ Clustered}{Number\ of\ Total\ Images\ Clustered}$$

Recall is defined as the number of relevant images within a cluster divided by all images within a cluster. It can be formally defined as

$$R = \frac{Number\ Of\ Relevant\ Images\ Retrieved}{Number\ of\ Total\ Relevant\ Images\ in\ Dataset}$$

Precision and recall provide measurements for determining the percentage of clustered images that were correct(Precision) and the percentage of the total relevant images that were clustered(Recall). These two values, however, trade off against one another. Whenever the precision of a cluster is quite high, the recall will generally be low and vice versa. It is difficult to know which of these two measures are more important for the purposes of this work. A high precision is desired, as inaccurate clusters will lead to noisy training sets and degrade the performance of machine learning classification approaches. A machine learning approach also requires a significant amount of data to create a robust model, therefore a high recall is also desired.

One evaluation metric that trades off precision against recall is the F-Measure, also called the F1 score [van Rijsbergen, 1979]. This is a weighted harmonic mean of precision and recall and provides a balanced evaluation metric. The F-Measure is formally defined as:

$$F = \frac{1 + b^2}{\frac{b^2}{P} + \frac{1}{R}}$$

where b is a weighted value that determines the importance of precision and recall. A value of $b < 1$ will emphasise importance on precision, where as a value of $b > 1$ will emphasise recall. In this work the value assigned to b is 1, which calculates the harmonic mean of precision and recall. To determine the F-measure metric, the following formula is used:

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R}$$

Contextual-Based Clustering

The first hierarchical based approach evaluated consists of contextual information only. Firstly, spatial based clustering was carried out at the top level of the tree, followed by contextual based clustering using the textual features that accompany

each image within the training collection using the methods outlined in section 4.3.2. For each of the chosen 250 seeds, all images within a spatial radius of 250 metres were clustered. At the second stage of the clustering pipeline, the textual features associated with all images were pre-processed and evaluated using three different metrics and a variety of thresholds. For each threshold, images were discarded from the cluster if the metric distance was above the threshold. The results of each cluster were then compared against the benchmark standard to ascertain the usefulness of textual information in clustering community contributed image collections.

As the 3 examined metrics measure correlation between sets of tokens, and 1 is the minimum score possible for all 3, a score of 1 will determines the minimum correlation possible between cluster images (i.e. Any score below 1 indicates that at least 1 match was calculated between the two sets of tokens, while any score of 1 determines that there was no match at all).

As can be seen from the results, one poor performance attribute from the perspective of this work, is the precision score. With regards to recall, clustering visually similar images based on tag similarities performs well for the purposes required in this work. At the lowest possible threshold value of 1, however, the highest recall score across all text based approaches is .42. This means that in utilising the text based features as part of a clustering algorithm, over 50% of all candidate images would be disregarded, irrespective of any threshold value chosen.

It would seem evident based on the high proliferation of tags describing a location, that in the absence of geo-tags, this contextual information could prove useful in the initial clustering stages of an hierarchical algorithm. Based on the results of Tables 4.2, 4.3 and 4.4, spatial information, provides a more accurate and cheaper comparison method with a higher level of recall than that of text

Threshold Value	0.6	0.7	0.8	0.9	1.0
Recall	.117	.179	.210	.301	.420
Precision	.085	.091	.067	.044	.028
F-Score	.098	.120	.101	.076	.052

Table 4.2: Textual Based Clustering - Jaccard Distance

Threshold Value	0.6	0.7	0.8	0.9	1.0
Recall	.210	.239	.301	.382	.420
Precision	.067	.058	.044	.030	.028
F-Score	.101	.093	.076	.055	.052

Table 4.3: Textual Based Clustering - Dice Coefficient Distance

Threshold Value	0.6	0.7	0.8	0.9	1.0
Recall	.277	.322	.360	.406	.420
Precision	.057	.045	.040	.031	.031
F-Score	.094	.078	.072	.057	.057

Table 4.4: Textual Based Clustering - Overlap Distance

based data, therefore textual based features were not included in the optimal HKM algorithm.

Single Low-Level Features

In this section, clustering was evaluated using solely one low-level image feature at the second stage of the hierarchical clustering algorithm. Four low-level features were evaluated; Colour Correlogram, Gabor Texture Feature (36 bin), MPEG7 Scalable Colour Feature and the MPEG7 Edge Histogram. Each feature was first evaluated on its own as a second level in the hierarchical k-means tree, with the first level being spatial based clustering as defined in section 4.3.1.

From the results in Tables 4.5 - 4.8 and Figure 4.8, it is evident that the best performing low-level feature for the purposes of this work is the MPEG7 edge histogram descriptor. The EHD provides a high precision score when using a low

threshold value. Using the two strictest thresholds (15 and 20) the EHD provides a F-score measurement of over .1. As expected, the colour features performed poorly in comparison to the EHD. It is assumed that this is because of issues with invariance to illumination changes. The scalable colour feature scored highly with regards to precision when used with a strict threshold value, however the corresponding recall score was very low and not sufficient enough for use in this work. The Gabor wavelet feature also performed poorly, this may be because the feature contains no geometrical information.

Based on these results, the MPEG7 feature was selected for use in other tasks described in Chapter 5, however it was not selected to be used as part of a clustering algorithm as it did not perform as well as more advanced image patch features, presented in Table 4.13.

Fusion of Low-Level Features

In this section, an hierarchical approach consisting of three levels was analysed and evaluated. This top level of the clustering tree consisted of spatial based clustering. The next layer consisted of clustering based on a global low-level image feature, followed by the final layer consisting of a second global low-level feature. The aim is to ascertain whether combined with geographical clustering, which already would have drastically reduced the feature space, if the combination of multiple low-level features is sufficient enough to produce accurate groupings of images. Four combinations of low-level features were evaluated.

- Spatial Colour + Texture
- Colour + Texture
- Colour + Edge

Threshold Value	1.0	2.0	3.0	4.0	5.0
Recall	.093	.222	.527	.819	.847
Precision	.061	.058	.037	.026	.024
F-Score	.074	.092	.069	.051	.048

Table 4.5: Spatial + Colour - Colour AutoCorrelogram

Threshold Value	10.0	20.0	30.0	40.0	50.0
Recall	.020	.055	.123	.196	.296
Precision	.060	.075	.071	.056	.045
F-Score	.031	.064	.090	.087	.078

Table 4.6: Spatial + Colour - MPEG7 Scalable Colour Feature

Threshold Value	1.0	3.0	5.0	10.0	20.0
Recall	.051	.124	.183	.270	.358
Precision	.034	.037	.035	.030	.024
F-Score	.041	.057	.059	.054	.045

Table 4.7: Spatial + Texture - Gabor Wavelets

Threshold Value	15	20	25	30	35
Recall	.125	.429	.694	.796	.814
Precision	.159	.057	.031	.025	.024
F-Score	.140	.101	.060	.049	.048

Table 4.8: Spatial + Texture (Edge Based) - MPEG7 Edge Histogram

- Texture + Edge

The results of this evaluation can be seen in Tables 4.5 - 4.8. From these results, it is evident that the combinations of low-level features actually degrade clustering accuracy. In the best performing fusion approach (Texture + Edge), the F-Measure score decreases by .006 when comparing against the score output by using the edge based feature on its own. The main reason for degradation across all fusion results is the poor performance of all low-level features excluding the MPEG7 edge histogram.

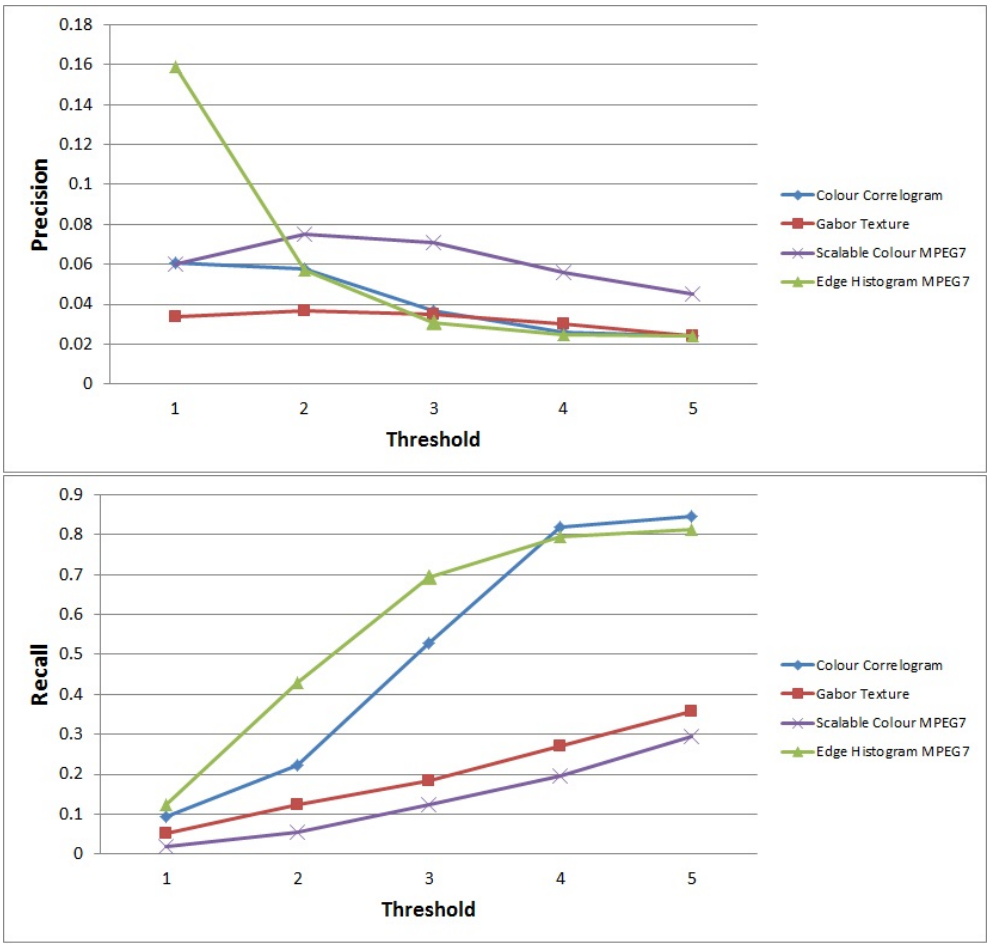

Figure 4.8: An illustration displaying the results of four clustering runs compared against the benchmark clustering results, using just one low level feature per run. The results in the top diagram are the precision values and the bottom diagram displays the recall values.

Overall, the fusion of low-level feature approaches performed poorly and this approach was disregarded for the remainder of this work.

Fusion of Local Image Feature Clustering

In this section, a hierarchical clustering algorithm consisting of spatial data, VBOW features and a graph based SURF correspondence measure was evaluated. In the visual word stage of the algorithm, 4 different values for k were analysed where k

Threshold Value	.75	1.0	1.25	1.5	1.75
Recall	.358	.368	.369	.369	.369
Precision	.021	.021	.021	.021	.021
F-Score	.040	.040	.040	.040	.040

Table 4.9: Spatial + Colour AutoCorrelogram + Gabor Texture

Threshold Value	.75	1.0	1.25	1.5	1.75
Recall	.145	.363	.629	.758	.804
Precision	.124	.062	.035	.026	.025
F-Score	.134	.107	.067	.051	.048

Table 4.10: Spatial + Colour AutoCorrelogram + Mpeg7 Edge Histogram

Threshold Value	.75	1.0	1.25	1.5	1.75
Recall	.128	.187	.284	.389	.389
Precision	.042	.034	.027	.021	.021
F-Score	.064	.058	.050	.039	.039

Table 4.11: Spatial + Colour AutoCorrelogram + Gabor Texture

Threshold Value	.75	1.0	1.25	1.5	1.75
Recall	.208	.344	.369	.369	.369
Precision	.040	.023	.021	.021	.021
F-Score	.068	.043	.040	.040	.040

Table 4.12: Spatial + Texture (Edge Based) - MPEG7 Edge Histogram + Gabor Texture

determined how many ranked images to retain in each cluster before expensive SURF based verification was carried out. These values were 50, 100, 150 and 200. The results of this evaluation can be seen in Table 4.13.

From the results in Table 4.13, it is evident that the fusion of spatial data, inverted visual word features and a graph based SURF verification process, performed well. This approach achieved a F-Measure score above any of the other evaluated methods, which is illustrated in Figure 4.9. The clustered images output by this approach for a random test image are displayed in Figure 4.11.

Top k Matches Threshold	k =50	k =100	k =150	k =**200**
Precision	.460	.462	.460	**.451**
Recall	.323	.378	.398	**.409**
F-Score	.380	.416	.427	**.429**

Table 4.13: Inverted Visual Words + SURF Geometric Consistency Matching

The performance of the approach increases as the value for k increases, however it must be noted that the processing time required also increases. In this work, the clustering algorithm with highest evaluated value for k required on average 51 seconds for each test cluster seed. While this was deemed acceptable for the purposes of this work as the processing is carried out offline, it must be taken into account when using a larger scale image corpus.

From informal empirical inspection, the algorithm seems to achieve the desired aims. For each cluster, the majority of images seem to be relevant and share similar viewpoints of a landmark. This is illustrated for a random test image in Figure 4.11.

Locality Sensitive Hashing

As described in section 4.4, locality Sensitive Hashing is an efficient method to find approximate nearest neighbours in high dimensional space. In other work, approaches have been proposed to populate LSH tables with interest point descriptors and cluster imagery based on nearest neighbours, however this method has a high memory footprint. Due to memory constraints, the input features evaluated in this work were global-based features, visual BOW histogram features with various vocabulary lengths.

When constructing the visual word histograms, 4 values for k were evaluated, 1000, 2500, 5000 and 10,000. Additionally, 4 values for C were also evaluated. The results of this evaluation can be seen in Tables 4.14 - 4.17.

From the results of the evaluation, it can be seen that the hierarchical approach using spatial data, visual bag of words features and a graph based SURF correspondence process, significantly outperforms the LSH based approach in terms of a balance between precision and recall. The LSH algorithm achieves a higher precision score when utilising a low value for C, however, using this parameter reduces recall to .01. This is not a suitable level of recall for the purposes of this work. Based on this result, it would appear that the locality sensitive hashing approach would perform well in a recognition task where a high level of precision is desired. When using a value of C that achieves a better balance between precision and recall, the f-measure score is well below that of the hierarchical approach. This is illustrated in Figure 4.10.

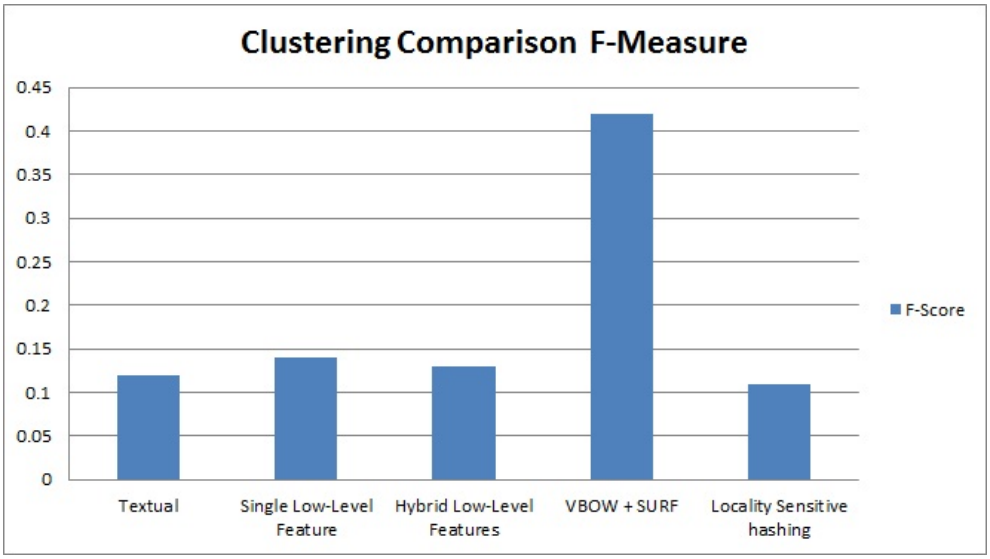

Figure 4.9: An illustration comparing the F-Measure scores for all evaluated clustering approaches

Visual Vocabulary Size	k =1000	k =2500	k =5000	k =10000
Precision	.595	.475	.363	.420
Recall	.018	.022	.031	.027
F-Score	.034	.042	.040	.050

Table 4.14: Locality Sensitive Hashing (C = 16) - Global Visual BOW Features

Visual Vocabulary Size	k =1000	k =2500	k =5000	k =10000
Precision	.161	.142	.246	.113
Recall	.083	.075	.046	.116
F-Score	.109	.098	.077	.114

Table 4.15: Locality Sensitive Hashing (C = 25) - Global Visual BOW Features

Visual Vocabulary Size	k =1000	k =2500	k =5000	k =10000
Precision	.143	.143	.115	.207
Recall	.085	.090	.093	.058
F-Score	.106	.115	.102	.090

Table 4.16: Locality Sensitive Hashing (C = 45) - Global Visual BOW Features

Visual Vocabulary Size	k =1000	k =2500	k =5000	k =10000
Precision	.211	.150	.096	.120
Recall	.055	.067	.136	.082
F-Score	.087	.092	.112	.097

Table 4.17: Locality Sensitive Hashing (C = 60) - Global Visual BOW Features

4.6 Conclusions

In this chapter, several methodologies were proposed to solve the problem of accurately clustering large numbers of community contributed images into visually near identical clusters. All of the clustering approaches implemented used spatial based clustering as the first stage in the clustering tree. The spatial clustering allowed for the significant reduction of the corpus before more expensive image comparison methods were processed. Without this spatial data, it is assumed that the accuracy of the clustering algorithms would decrease due to the increased

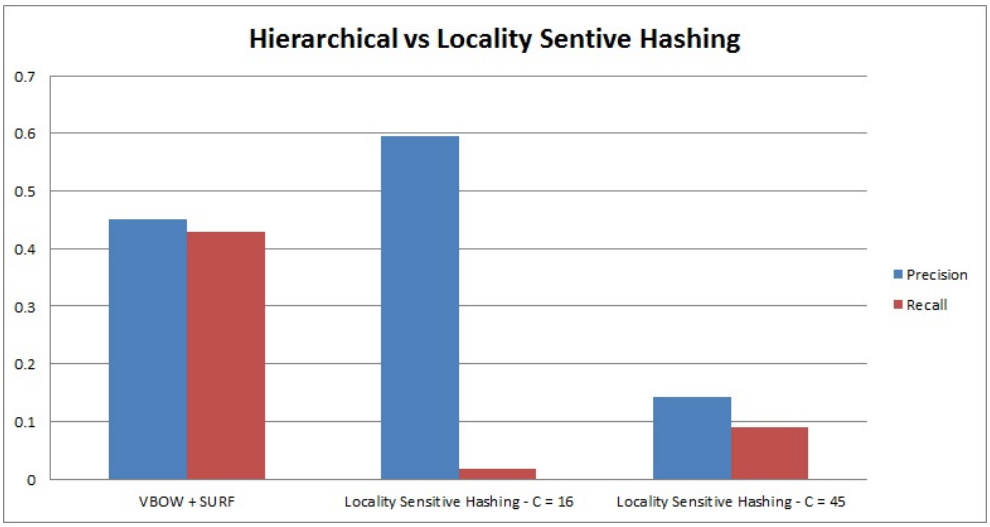

Figure 4.10: An illustration displaying the results of four clustering runs compared against the benchmark clustering results, using just one low level feature per run. The results in the top diagram are the precision values and the bottom diagram displays the recall values.

size of the clustering space. The discrimination value of the image features used in this chapter decreases as the size of the image corpus increases.

Text based features were evaluated and produced some encouraging results but a limit on the recall score meant that they were not utilised. It is however assumed that in a landmark recognition system where geographical information is not available, text based features would perform well as a first stage in a clustering algorithm.

Low-level image features were shown to perform poorly with the exception of the EHD feature. The fusion of multiple low-level features surprisingly hindered clustering performance. Overall, it has been shown that these features lack the discrimination power to accurately cluster images into visually near identical viewpoints of landmarks.

In the evaluation section 4.5.2, it has been shown that a hierarchical approach based on VBOW features, followed by a graph based SURF correspondence process, outperformed all other evaluated clustering algorithms. This approach achieved an F-Measure score of .429 with a k parameter of 200. Through the

use of spatial data and a first stage clustering process consisting of visual word matching, the corpus was significantly reduced for each test cluster centre and allowed for the clustering algorithm to process each test image in just over 44 seconds on average. This is a significant improvement over using a brute force approach, such as that used in the benchmark, which required over 1 hour per test image.

Based on the performance of this approach, it was selected to be used as the clustering algorithm to group images in the training phase of the machine learning models used at the core of the landmark recognition framework proposed in this thesis.

Figure 4.11: A visualisation of the top 12 ranked clustering results and the bottom 12 ranked results for a randomly selected cluster seed in the clustering evaluation set. The total size of the cluster was 78. This clustering was based on the optimal hierarchical k-means approach that was selected to use for the remainder of this work.

Chapter 5

Landmark Recognition with Computational Classification Techniques

5.1 Introduction

In this chapter, the hypothesis that SVM classification models can be used as part of robust methods to classify an individual landmark within an image is explored. The chapter begins with some motivation behind this approach and a description of the proposed framework. A background of machine learning is provided along with a description of the two machine learning algorithms used in this work. An introduction to each of the image features used in this framework is then provided. The chapter concludes with an evaluation of all proposed approaches.

There are three main advantages to be gained from the use of classification models over other approaches:

- Computational overhead: The amount of time taken to compare and classify images in a large-scale database is significantly reduced. With efficient

filtering methods, this classification could be done in near real time in large-scale databases.

- Memory Requirements: This approach does not require that large amounts of data need to be stored in heap memory at classification time, therefore, there is no restriction on the maximum size of the image corpus.

- Robustness: Increased robustness is obtained by combining features obtained under multiple imaging conditions into a single model view.

The main motivation behind this approach is that in the tourist districts of many world cities, there tends to be a large proliferation of photos taken of the landmarks situated in these districts. Traditional approaches to landmark classification filter a dataset using geographical data and the carry out local image feature matching on the pruned dataset [Qamra and Chang, 2008]. The problem with this approach is that in the popular tourist areas of a city, there can be a very large number of images in the filtered data (even with spatial filtering). A large number of these images will overlap visually as tourists tend to take images of the same viewpoints of landmarks from similar perspectives. The aim of this approach is to take advantage of this overlap by creating a classification model to recognise these clusters of visually similar images, thus significantly reducing the time required to match a test image to a landmark or a certain viewpoint of a landmark.

One advantage of using machine learning models is memory requirements. One commonly used approach to landmark classification and retrieval is the use of large visual bag of words vocabularies combined with inverted index files [Philbin et al., 2007]. This work reported that this technique works very efficiently, with typical queries to an image dataset numbering over one hundred thousand images processed in just 100 milliseconds. They organise the inverted index into what they describe as a 'space efficient binary-packed structure'. This structure

is then loaded into the main memory of a system. The main problem with this approach is that memory constraints become an issue once the image dataset is scaled up. The memory required to store the inverted index of an image set of just over one million images is 4.3 gigabytes. This is larger than the maximum addressable memory that is available in a 32-bit machine.

Another approach to landmark classification is to organise local image features into different types of tree structures (Kd-Tree, hierarchical tree) and match test image features by proliferating them down the tree [Yan-Tao Zheng, 2009], [Schindler et al., 2007]. These tree structures require significant building time and must be rebuilt each time a classification system is re-run. The structure itself is loaded into heap memory and is thus subject to memory constraints.

A single SIFT feature vector has a length of 128. Each entry within this vector contains a float value, which is 4 bytes is size. This means that to store a single SIFT feature requires 512 bytes of memory. The maximum addressable memory in a 32-bit system is 4 gigabytes. Therefore, the maximum number of interest point vectors (without additional information such as scale, indexing values or spatial information) that can be stored in a vocabulary tree on a standard 32-bit machine is around 8 million.

In the approach proposed in this chapter, the memory requirement is static. Machine learning models can be stored in hard disk memory. The memory footprint of each model is quite small, can be loaded into physical memory quickly, and models do not have to be loaded into memory at runtime. The means that the only limitation to the size of the dataset is hard disk space. This represents a considerable advantage over many of the other approaches proposed in the literature in terms of scalability.

5.1.1 System Overview

In this section, a brief introduction into the process of classifying a test image using machine learning algorithms is provided. The first stage of the system consists of grouping the entire training corpus into clusters of near identical imagery. This clustering process is described in Chapter 4. For a specific spatial region, all clusters within that region are then input into a machine learning algorithm with each cluster representing a class. A classification model is then trained for each region consisting of these classes. For each model, a spatial location is assigned to it based on the mean location of all images within the model. This process is illustrated in Figure 5.1. All of these spatial coordinates are then saved into a database. Each classification model is a multi-class classification model. This is a model that does not make a binary decision, such as whether the image is associated with a specific class or not, but rather outputs a nearest neighbour class label from a finite number of classes used to train the model. This is illustrated in Figure 5.2.

To classify a test image using the system, firstly, the location information associated with that test image is firstly extracted. This information is used to retrieve the nearest model from the system using the closest distance between geographical coordinates. Image features from the test image are then extracted and processed through the model. All images associated with the outputted class/cluster from the machine learning algorithm are then analysed and ranked according to the number of interest point correspondences (using the SURF algorithm) between them and the test image. If the number of correspondences is above a threshold of 3, an image is considered a match. This classification process is illustrated in Figure 5.3.

Throughout this chapter, all of the evaluated classification approaches are based on algorithms that form part of a facet of computer science called machine learning. In the next section, a brief introduction to this field is provided.

5.2 Machine Learning

Machine learning grew from early research in the field of artificial intelligence and is now widely used in applications across many research fields, such as predicting stock prices [Huang et al., 2005], classifying protein types from DNA sequences [Ma et al., 2009] and classifying semantics from image content [Rafiee and Sarajian, 2008]. Machine learning was described by Arthur Samuel as 'The field of study that gives computers the ability to learn without being explicitly programmed'. Machine learning methods are typically used in situations where there is no evident solution for a programming problem, and instead a computer 'learns' a desired output from a training set of inputs.

Consider the problem of trying to program a computer to recognise a handwritten character image. It is very difficult, if not impossible, to create a set of programming instructions to recognise a character from an image. There will be many discrepancies that arise from different people writing the same letter. There will be deviations in size and shape and there is no way to predict and describe how a letter will appear from person to person. However, there are many examples of handwritten images of characters available, along with their associated letter that they are meant to represent. One solution to this problem is for the computer to classify the output based on a set of these labelled training examples (This is the same method that children use to classify certain semantics, learning by example. For example, Children will learn to recognise that a room is untidy by being shown rooms in untidy states and being told that they are untidy, rather than be given a strict definition of the word untidy). A collection

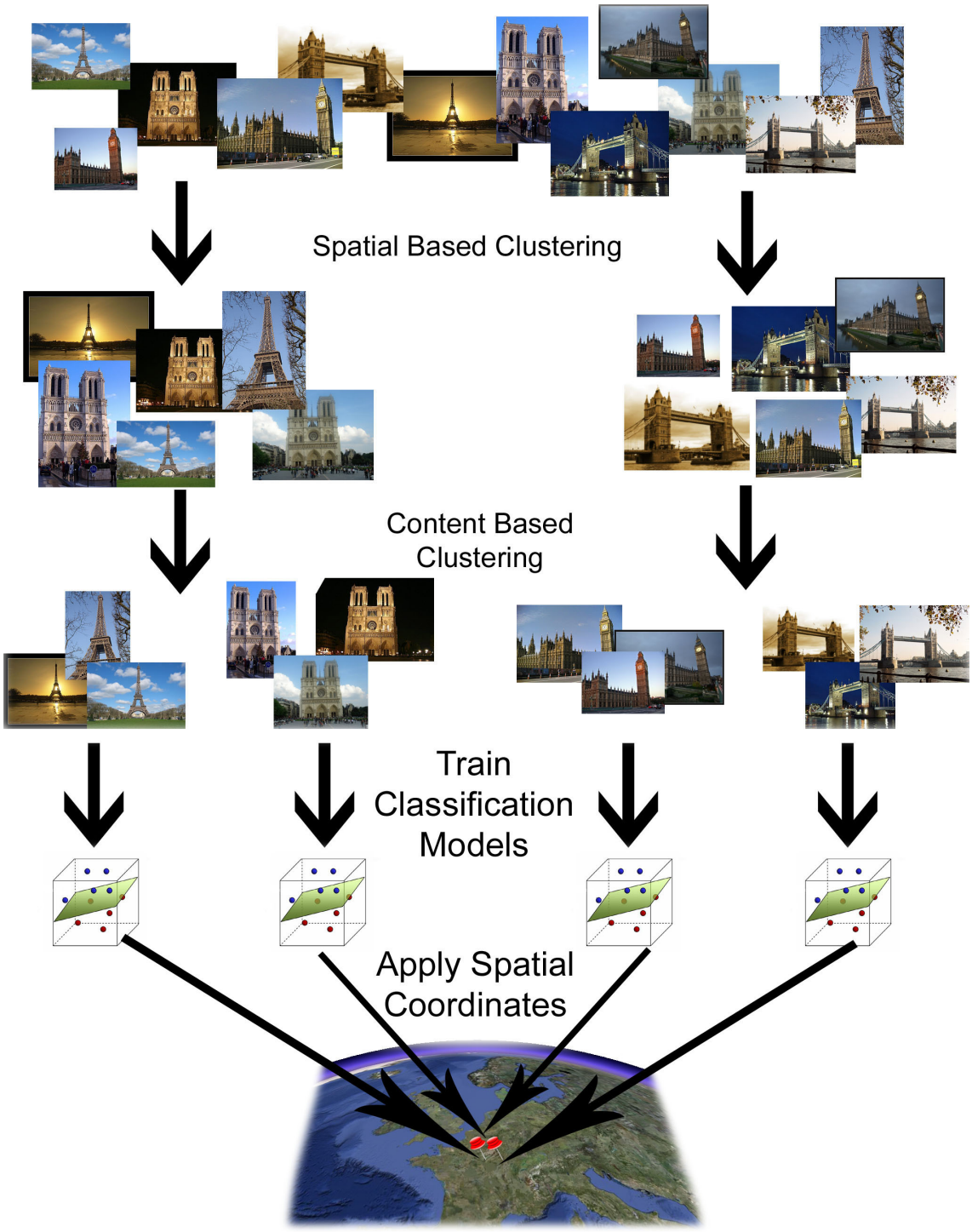

Figure 5.1: An illustration of the SVM training process using a small set of data. Images are clustered based on geographical location. These clusters are then subclustered based on image content.

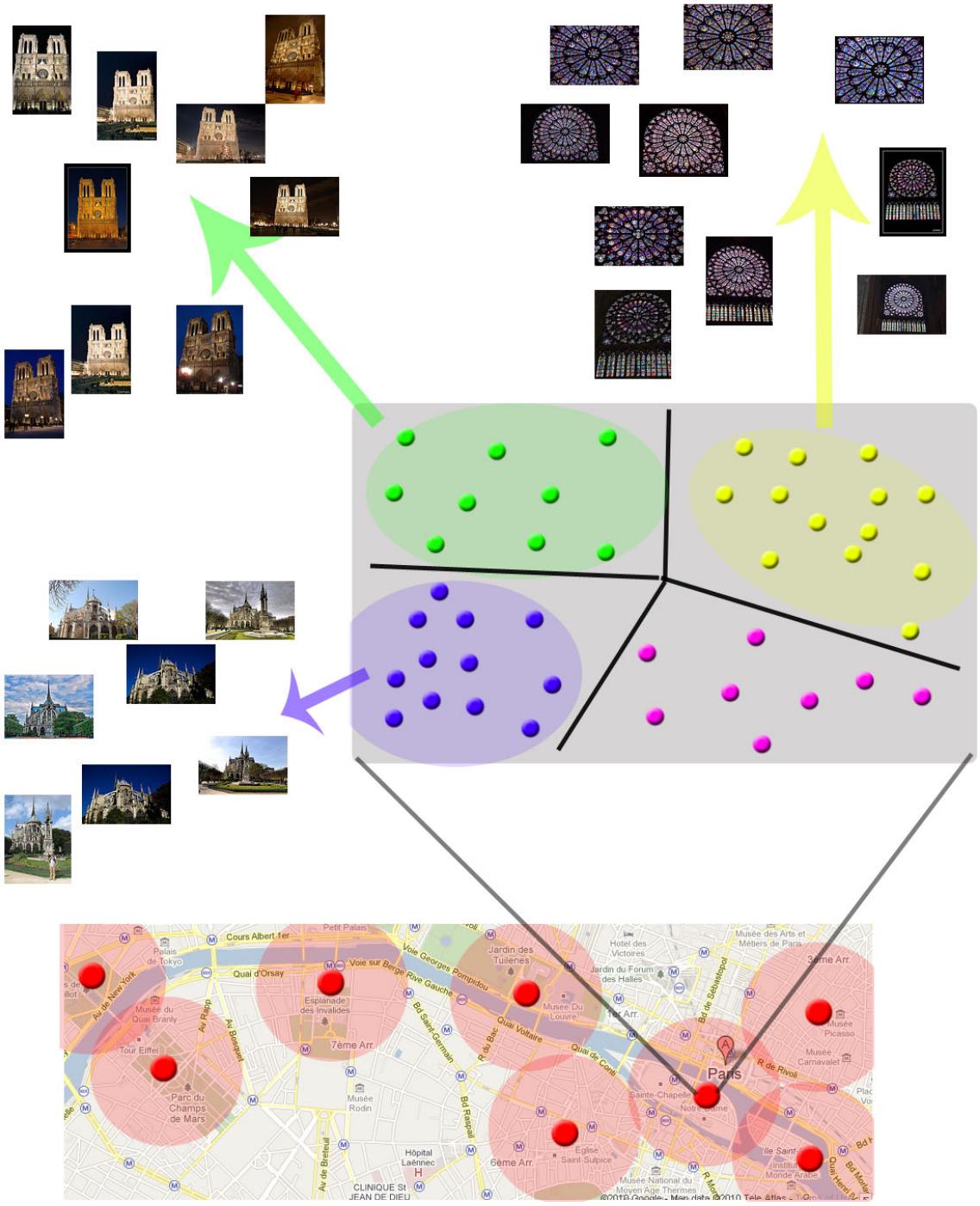

Figure 5.2: A diagram illustrating how the multi-class SVMs are trained and spatially organised. Here is an example from the centre of Paris where 8 multi-class SVMs have been trained to recognise a number of landmarks from different viewpoints at different spatial locations. The light red circles around each classification model centre represent the spatial radii used to select clusters to include in the model.

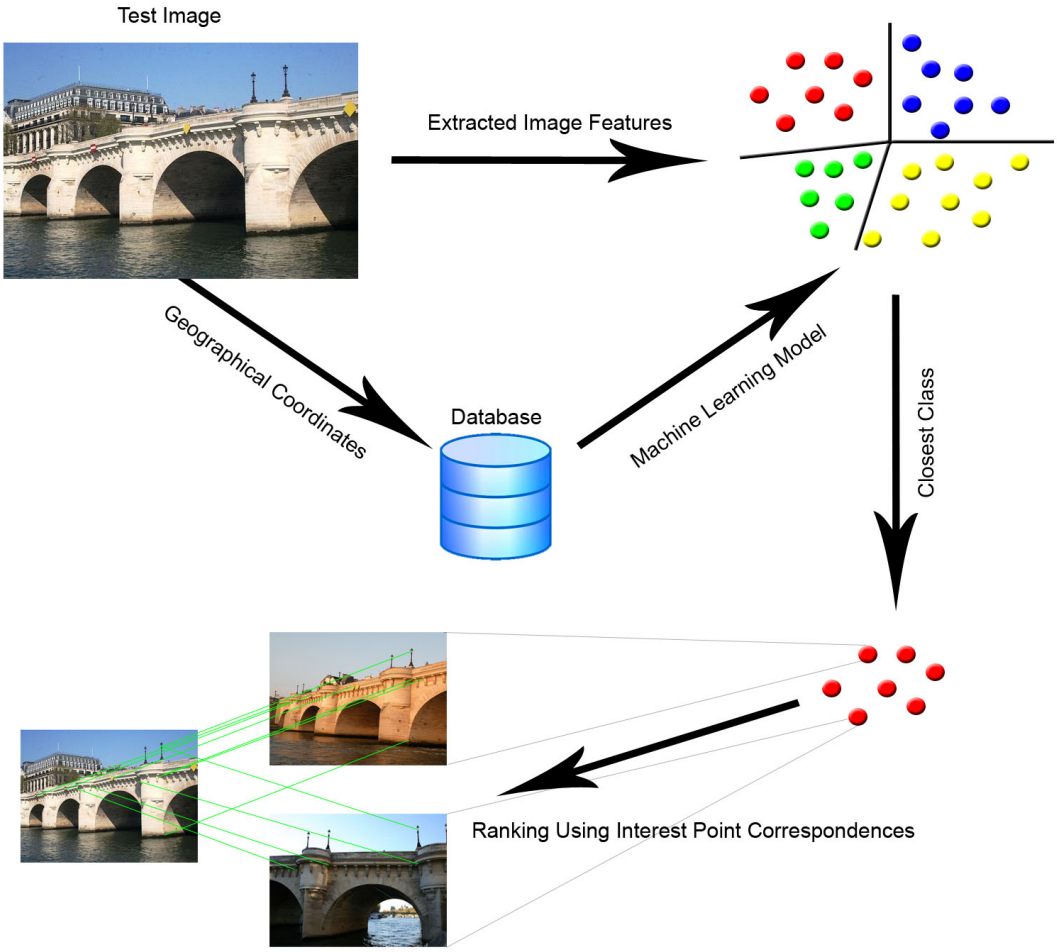

Figure 5.3: An illustration of the process of classifying a test image using machine learning classification models.

of positive and negative examples are collected and using learning algorithms, a model is created to recognise input images with similar patterns to the positive sample data. An input image is then compared against this model and a machine learning algorithm will try to predict the similarity of the input image to the sample images. Machine learning methods can be roughly divided into two main types:

- **Supervised Learning** Supervised learning is the process of creating a classifier based on a set of inputs that is created by a human supervisor. Given a set of inputs (x_1, x_2, x_3...) and a set of desired outputs y_1, y_2, y_3...., the aim of a supervised learning algorithm is to produce the correct output when given a new input. This process of classifying data that is not located in the training examples is called generalisation.

  There are three main types of supervised learning:

  1. **Binary Classification** is when the outputs from a machine learning algorithm will be binary. An input example belongs to a class or not. An example of binary classification is face detection in images, where the outputs are either true (image pattern was classified as containing a face) or false (image pattern was classified as not containing a face).
  2. **Multi-class classification** is where an output will be classified into one of a finite number of classes. A training set of input features will contain a number of different class labels and the classification algorithm will output the label with the highest prediction score. In the case of SVMs, a multi-class classifier is an extension to a binary classifier consisting of n-1 binary SVMs where n is the number of class labels in the training set.
  3. **Regression** is where the desired output is not a member of a set class, but rather a predicted value based on a set of statistical inputs. The machine learning algorithm would learn the relationship between a sample set of statistics and associated output values and then attempt to predict an output value based on a similar set of statistical data. One example of where regression is used is in the forecasting of stock market prices [Ping-Feng and Chih-Sheng, 2005].

- **Unsupervised Learning** In supervised learning, the process of a machine learning algorithm is to make a prediction based on a set of inputs that are provided by a 'supervisor'. In unsupervised learning the input data is unorganised. The aim of unsupervised learning is to find structure or similarity in the data, and without human supervision or the aid of a fitness function, organise it into structured inputs. One method for this automatic structuring of data is called clustering, which is described in Chapter 4. Clustering involves grouping sets of data based on the distances between features using a similarity measure. Several well known algorithms exist to cluster data with the most commonly used being the K-Means algorithm [MacQueen, 1967].

5.2.1 Anatomy of a Machine Learning Algorithm

In machine learning, there is typically a relationship between a set of input features ($x_1, x_2, x_3...$) and the outputs produced $y_1, y_2, y_3....$ This relationship is built on a target function, and it is this function that a machine learning algorithm seeks to replicate. The actual estimate of this function that is produced by a machine learning algorithm is called a decision function. This decision function is selected from a set of possible functions which map the input features to the produced outputs. This set of possible functions are known for historical reasons as hypotheses. The algorithm that analyses the sets of input features and selects a hypothesis function from the set of possible hypotheses is referred to as the learning algorithm.

The ability of a machine learning algorithm to learn a training set without error is defined as the capacity of the machine. An algorithm with a high capacity would be able to learn a large number of input/output parings irrespective of how they are labelled. A hypothesis that can learn a training set without large

amounts of error is said to be consistent. An algorithm with a high capacity does not necessarily mean that the algorithm will perform well. For example, if a training set is noisy, there is no guarantee that the hypothesis will correctly map the input/output function.

The aim of a learning algorithm is to classify feature vectors that are not present within the training set. This is called generalisation. It is common that many machine learning algorithms will be able to correctly learn the training set, but may produce random predictions on any test data not in the training set. This phenomenon is known as overfitting and can lead to poor generalisation performance. Overfitting can commonly occur due to too much complexity in the decision function. One approach to overcome overfitting issues is to ensure that an algorithm's capacity is kept low. If an hypothesis' capacity is too low however, many important patterns in the training set will be ignored. Therefore it is important that the capacity of a hypothesis is kept balanced. Two commonly used machine learning algorithms are used as part of this framework. In the next section, a brief description to both of these algorithms is provided.

5.2.2 Commonly Used Machine Learning Algorithms

Nearest Neighbour Classification

One of the most basic classification methods is nearest neighbour (NN)classification, which is a supervised learning technique, and although basic, it can be a very effective classification method. NN classification is based on the similarity between an unlabelled sample feature and the closest feature to it in a labelled training set. The label associated with this closest feature is then applied to the sample. The distance between the sample feature and all features in the training set is usually calculated using an established distance metric such as the Euclidean distance. Although nearest neighbour classification can be accurate in certain cases, bas-

ing classification on a single nearest neighbour can be particularly sensitive to outliers in training sets. K-Nearest neighbour classification address this issue by calculating the nearest class of features based on a number of nearest neighbours in feature space. K-nearest neighbour (k-NN) classification is based on a majority vote among the k closest neighbours in feature space, where k is a positive integer value and all neighbours have a known class value associated with them.

The algorithm can be described as follows: Given a training collection of n labelled image features $T = v_1, v_2....., v_n$, and a k value of 1, the algorithm will assign to a test image I_{test}, from which an image feature v_{test} is extracted, the label associated with its closest neighbour in T based on a distance measure such as the Euclidean distance or the Mahalanobis distance. If the value of k is higher than 1, the test image I_{test} is classified as belonging to the label that has the majority vote within the k nearest neighbours in T. The higher the value of k, the more robust the classifier is to noisy data, but more processing time is involved in classification. This issue is illustrated in Figure 5.4.

K-Nearest Neighbour classification is a simple model, but can often provide accurate prediction results. Due to the simplicity of the model, there are drawbacks however. The K-NN classifier is prone to errors, particularly with outlier features and noisy data. Therefore, it is necessary to evaluate and select an optimal value for k to avoid this. The other machine learning algorithm used in this work is called Support Vector Machines (SVM).

5.2.3 Support Vector Machines

A SVM is a learning algorithm originally developed by Vapnik [Cortes and Vapnik, 1995] that can perform input/output mappings from labelled examples and can choose a balanced capacity for each decision function. SVMs have been widely used in

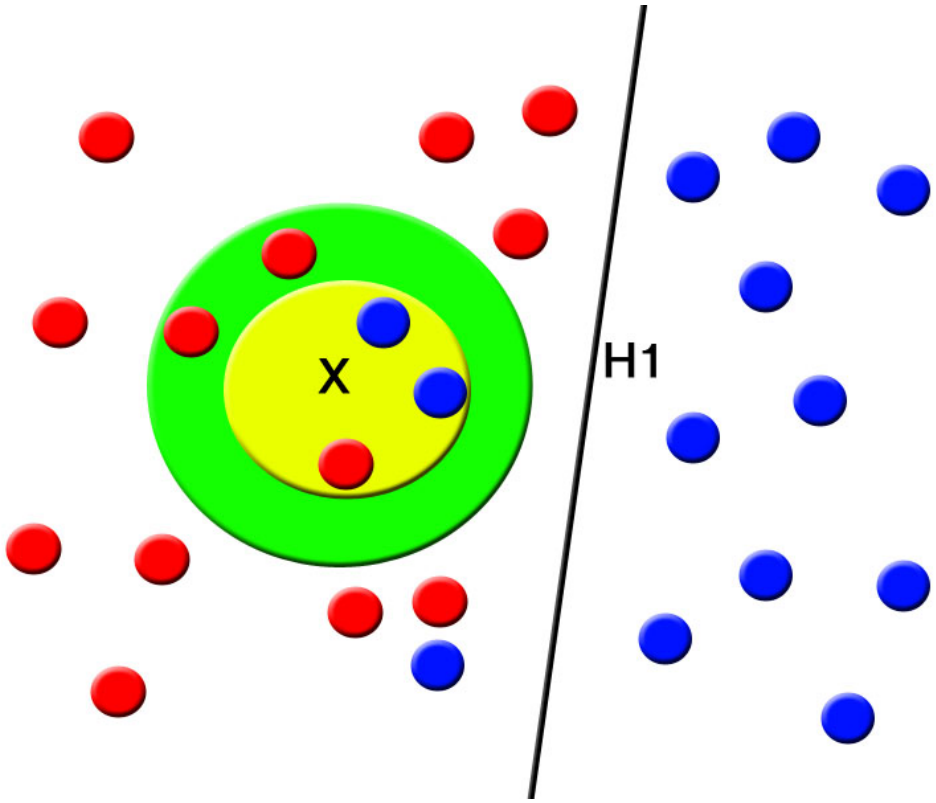

Figure 5.4: A diagram illustrating K-NN classification on a sample two class training set (class a = red, class b = blue). The value of k can play an important part in the accuracy of the classification. The hyperplane H1 represents a decision function that might be created using a linear classifier on this training set. From this hyperplane, it is evident that X should be classified as belonging to class a. In this toy example, if k is given a small value such as 1 or 3 (yellow circle), the test feature (X) will be mis-classified due to two outliers from class b being near to the test feature X in feature space. However if k is given a value of 5 (green circle) or larger, the test feature will be correctly classified as belonging the class a.

many different research genres and are highly regarded for scaling well with high dimensional data [Lin and Nevatia, 1998].

Applications of Support Vector Machines

In this work, SVMs were chosen as the machine learning algorithm for classification tasks, mainly because SVMs have been applied recently to a variety of real world problems. Of particular relevance in this work is that their performance has compared favourably to other machine learning approaches in the field of

image classification [Chapelle et al., 1999]. SVMs also scale well to high dimensional data and allow for a fast classification process, even with the use of high dimensional data.

Linear Classification

The main aim of an SVM is to seperate classes of data with the use of a hyperplane, an example of which is displayed in Figure 5.5. The general equation for a hyperplane H is

$$H = w \cdot x_i + b >= 1 \text{ where } y_i = +1$$

and

$$H = w \cdot x_i + b <= -1 \text{ where } y_i = -1$$

where x is an input point (a vector) lying on the hyperplane, w is a set of weights (also a vector) and b is a constant. H_1 and H_2 are two hyperplanes, that are parellel to H where

$$H_1 = w \cdot x + b = 1$$

and

$$H_2 = w \cdot x + b = -1$$

The points that lie along the hyperplanes H_1 and H_2 are the closest points to the the hyperplane H and are called the support vectors. The support vectors are the critical elements of the training set as they are the input features that would influence the position of the dividing hyperplane decision if removed from the dataset. Distance $d+$ is defined as the distance from H to the closest positive point, while distance $d-$ is defined as the distance from H to the closest negative point. The margin of the separating hyperplane is defined as $d-$ + $d+$. This margin can be calculated as $2/||w||$.

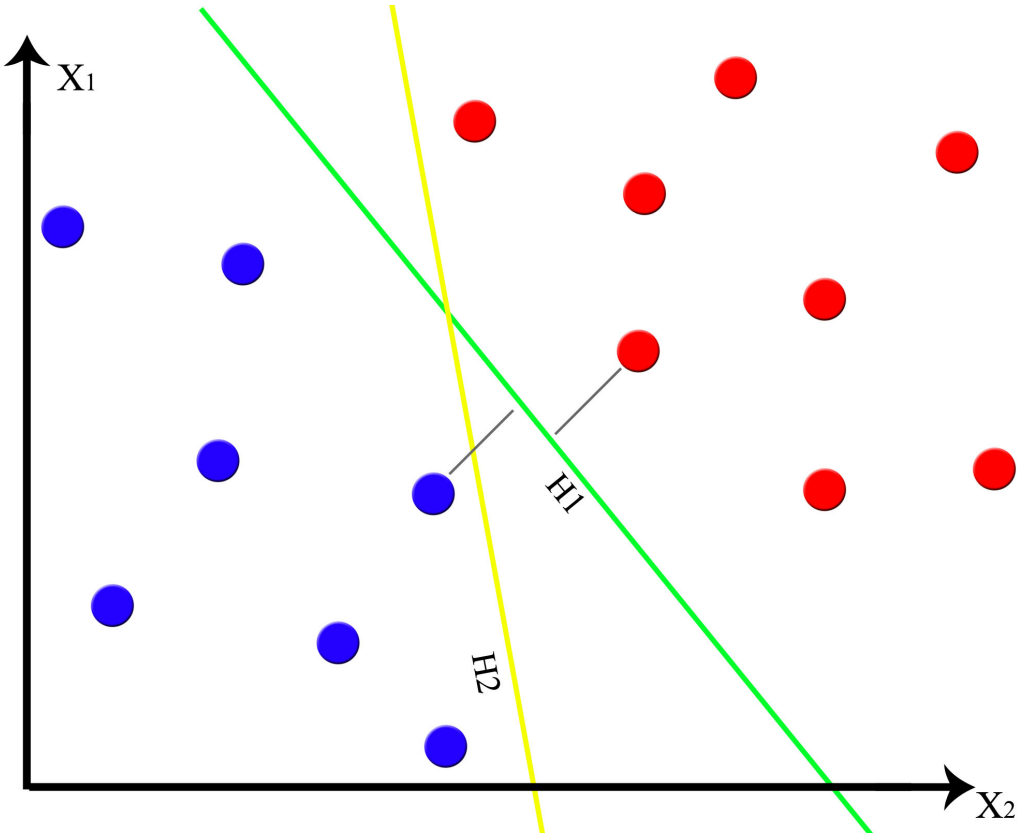

Figure 5.5: An illustration of two hyperplanes (H1 and H2) separating two classes containing two dimensional data. Hyperplane H1 separates the data with the maximum margin. H2 separates the data, but not with the maximum margin.

The main aim of SVMs is to create a hyperplane with as large a margin as possible, i.e. optimise w and b so that $2/||w||$ is maximised, which is the equivalent to minimising $\frac{1}{2}||w||^2$. A maximum margin hyperplane ensures a higher certainty level of correct classification, as points located near the decision plane represent unpredictable classification decisions. A classifier with a maximum margin will make much fewer of these low certainty decisions. This provides a slight margin of error within the classification procedure. A noisy variable will not cause a classification error.

Non-Linear Classification

When classifying data, it is possible that the dataset will be linearly separable, however, as the inputs become more complex, it is more likely that the datasets will not be linearly separable. This is particularly the case when dealing with large numbers of high dimensional image feature vectors. In most cases, even if the data could be separated linearly, it would be desirable to be able to separate the majority of the data while ignoring a small number of noisy input features (outliers). This ensures that these features would not significantly affect overall classification accuracy. Commonly a larger margin classifier can be created by allowing the classifier to misclassify some of these noisy inputs. This type of margin is called a soft margin classifier.

SVMs allow for soft margin classification with the use of slack variables, which allow an input within the margin to be misclassified. These slack variable constraints can be formally described as:

$$H = w \cdot x_i + b >= 1 - \xi_i \text{ where } y_i = +1$$

and

$$H = w \cdot x_i + b <= -1 + \xi_i \text{ where } y_i = -1$$

where the value ξ_i is a measure of how much a particular variable violates the original constraints. Any training feature with a value for ξ_i that is higher than 1 will be misclassified. Training features with a value of $0 > \xi_i < 1$, will be classified correctly, but will fall inside the margin. All other points in the model will have the value $\xi_i = 0$.

In soft margin SVMs a variable $C$is defined, which is a cost variable to discourage the use of slack variables ξ_i. If a large value is given to C, a large penalty is assigned to errors. The selection of a relevant C parameter when training an SVM model is then very important, as it instructs the decision function to prioritise

either maximising the margin (in the case of a small value for C) or minimising the amount of classification errors (in the case of a large value for C). The optimisation problem can then be formulated as minimising $\frac{1}{2}||w||^2 + C\sum_i \xi_i$

Kernel Functions

In many real-world applications, complex datasets might not be linearly separable and will require more complex hypothesis than linear functions. One approach to solve this problem is the use of multiple layers of linear functions, which in turn lead to the creation of multi-layer neural networks. SVMs solve this problem through the mapping of a feature vector into a different feature space using a simple mathematical function based on the inner product values between feature vectors.

In situations where complex data is not linearly separable, it might be possible that a transformation of this data into a higher dimensional space could result in a linearly separable model, where the linear based SVM approach described above could then be applied. The function behind this transformation, or mapping to a higher dimensional feature space is referred to in the literature as the kernel function.

For the duration of this work, an SVM classification library called libSVM [Chang and Lin, 2001] was utilised. There is no guarantee that a dataset will be separable in a higher dimensional space, and for different tasks, different types of mappings to different feature spaces will perform more accurately. For the purposes of this work it is therefore important to evaluate which kernels will perform best. In total, 4 kernel functions implemented in libSVM were analysed to evaluate their classification performance with the Paris corpus. These functions were:

Linear Kernel

The most basic kernel evaluated was the linear kernel. A linear kernel fuction relies on a dot product between two sets of high dimensional features. It can be formally defined as:

$$K(x, y) = x^T y$$

Polynomial Kernel

The polynomial kernel is a directional function which means the outputs depend on the directionality of the original low-dimensional input vectors. Due to this directionality dependence, it is assumed that the polynomial kernel might not perform as well as other kernel functions in one versus all, classification, as differences in directions of input vectors could mean that the data might not converge. The kernel function is defined as:

$$K(x, y) = s(x \cdot y + c)^p$$

Radial Basis Function Kernel

Possibly the most commonly used SVM kernel function is the Radial Basis Function. This function takes a parameter called gamma (g) that defines the influence of each support vector. A large gamma value will enable a support vector to have a stronger influence over a larger area, which in turn can lead to a smaller number of support vectors in each classifier. With stronger influence over larger areas, fewer support vectors are required to define a boundary. In libSVM, a default value of $\frac{1}{n}$ is assigned to the gamma parameter, where n is the number of input features in the model. In this work, optimal values for g were defined through *k-fold cross validation*. The RBF kernel is formally defined as:

$$K(x, y) = exp(-g||x - y||^2)$$

Sigmoid Kernel

The Sigmoid kernel, also known as the Hyperbolic Tangent kernel, is borrowed from another class of machine learning algorithms known as neural networks (NN). In neural networks, the Sigmoid function is often used as an activation function for neurons within a NN. An SVM classifier using the Sigmoid kernel is the equivalent of a two layer perceptron neural network. The kernel is formally defined as:

$$K(x, y) = tanh(kx \cdot y - \delta)$$

Parameter Validation

The majority of classification methods have one or more parameters that can be tweaked to improve or hinder classification accuracy, such as the k parameter in the k-NN classifier or the C value in SVMs that is associated with the cost applied to outliers in a training set. One important challenge that needs to be addressed is how to optimally select these free parameters to ensure a low generalisation error and avoid overfitting. If there is access to a large number of example images, this problem is trivial to solve. All examples can be processed through the classifier with each of the possible parameters (or a subset of the parameters) assessed, choosing the set with the highest classification rate or the lowest error rate. In practice however, it is more likely that there will rarely be an excess of training images and therefore an alternative method is desired.

One alternative approach to validating parameters is to extract examples from the training set and create a model minus these examples. The removed examples are then considered test images and classified by the model to ascertain performance. This approach to parameter optimisation is called cross validation and is used extensively in this work. Every classification model trained as part

of this investigation has its parameters selected using a type of cross validation called *k-fold cross validation*.

K-fold cross validation is a technique where the training set is randomly split into k disjoint sub sets of equal sizes (n/k where n is the total size of the training collection). A model is trained k times, with a separate sub set excluded each time and used as a test set (therefore model consists of $n-k$ training examples). Each of these k test sets are then classified against the model and the overall performance of that model is calculated as the mean accuracy of the k classification runs. In this work, a value of 5 is chosen for k, which is empirically determined to provide a reasonable balance between processing time required to train each model and the accuracy of the validation.

5.2.4 Multi-Class Support Vector Machines

SVMs are fundamentally binary classifiers, which are classifiers that are used to predict between two classes of data or whether an input belongs to a class or not (e.g. positive or negative output). It is possible, however, to extend a support vector machine to enable it to classify between multiple classes. These types of SVMs are called multi-class SVMs.

One common implementation of a multi-class SVM is called the 'one versus all' method. This is a technique where *k* SVM models are trained, where *k* is equal to the number of classes. The *i*th model is trained with all sample inputs that are associated with the *i*th class, and all these inputs are labelled as positive. All other sample inputs are labelled as negative.

The technique adopted in this work uses the 'one vs one' method. This is an approach where $(k-1)$ binary classifiers are trained for each class. The *i*th model for each class is trained using negative inputs that are associated with the *i*th class.

A test feature is then classified against all of these models and the optimal class is defined using a voting approach.

5.2.5 Input Features for SVM Classification

As the number of interest points detected within an image using an algorithm such as SURF is dependent upon the salience of each individual image and not based on an algorithm parameter, the features are not considered static. As these features are not static, they are not suitable for use as inputs in SVM classification. SVM algorithms require that all input vectors are of identical length, and each feature is in the same vector position for each input. It is therefore necessary to quantise local features into a fixed length feature vector or to exploit global features.

Low-Level Image Features

Many low-level image features are global and consist of feature vectors comprising a fixed length, and are thus suitable as input features to classification models. The main issue with using low-level features is that they might not discriminate well enough to ensure a high degree of classification accuracy. Colour features will not be suitable for use in this classification task, due to the fact that they are sensitive to small illumination changes. Additionally, it might be necessary to classify a black and white image, using this framework, which would make colour based features redundant. Based on these considerations, the main features that were experimented with were based on image texture (MPEG-7 Edge Histogram) and local image features (SURF).

MPEG7 Edge Histogram

In Chapter 4, the MPEG7 edge histogram feature was shown to outperform the other low-level features analysed (Gabor Texture, Scalable Colour and Colour Auto-Correlogram) as part of an image clustering process. Based on these results, the first input feature evaluated for use with the machine learning based framework outlined in this chapter is the EHD (described in section 4.3.6).

Visual Bag Of Words

It is not possible to use interest point descriptor values as inputs into a machine learning algorithm as the feature space is not static. Interest point features are detected in salient regions of an image. Some images have more salient regions than others, therefore, different numbers of features will be extracted from different images. An approach is desired that will quantise all interest points within an image into a fixed length, global descriptor, so that it is possible to use interest point features with machine learning algorithms. One such approach is called the visual bag of words model (VBOW). Bag of words (BOW) models have been used in document classification successfully in the past [Lebanon et al., 2007] [Metzler, 2008] [Torkkola, 2002]. A BOW model is a technique where a document is represented as an unordered collection of words that are then used to classify a document based on these representations. VBOW features are based upon the same basic premise, however the bag of words is replaced by a bag of descriptions of image patches. These image patches can be identified from a sample set of images using a variety of approaches such as dense sampling, random sampling or using an interest point detection algorithm, such as SURF. Descriptor vectors are then processed for each of these image patches, usually using an established algorithm such as SIFT or SURF. A collection of these descriptors is referred to as a visual vocabulary or a codebook.

Once a codebook is created, the VBOW approach provides an efficient method to quantise large numbers of image descriptors. Each image is represented as a bag of visual words that are created based on the presence of visually similar image descriptions of salient regions in an image and contained within the visual vocabulary [Tirilly et al., 2008].

There are several steps involved in creating a VBOW model:

- Local image feature descriptions are extracted from each image or from a subset of images within the dataset.
- These image features are then quantised into a visual vocabulary using a k-means clustering algorithm, with k being the vocabulary size of the dictionary.
- Using this vocabulary, each image can then be represented by a global histogram value that is calculated by comparing each image feature to every feature in the dictionary and a vote is counted for the entry in the dictionary that has the smallest distance from the image feature. The histogram forms a vector where the number of possible words is the length of the feature vector.

This VBOW model effectively quantises large numbers of local image features into a single feature vector, while retaining a high level of discrimination which is illustrated in Figure 5.6. A VBOW histogram is an orderless image feature, in that the order of feature values is not determined in advance, and has little or no impact of classification/matching accuracy. These histograms are considered to be global image features as they represent all the content of an image. Due to this global representation, there are several drawbacks to the visual BOW model. There is no way to extract information solely about individual objects (i.e. landmarks in this work) or shape information describing these objects. This

means that the model could be capturing redundant data such as occluding objects or background features, for example, that might not be desirable to be matched. The process of extracting specific objects from an image is called object segmentation and this is still an active research area. No standard approach is known to work optimally for all objects, and therefore many approaches can often be unreliable and inaccurate. One alternative approach to object segmentation that has been suggested to overcome this object versus global representation, is suggested in [Xiao et al., 2010]. This approach however, relies on a lot of data and near duplicate images for each object/landmark to be available.

Even without this localised information, the VBOW model has been shown to work well in not only identifying a scene, but also for the classification of whether a specific object is present in a scene or not. Additionally, it could be argued that the lack of spatial information in the feature, while lacking discrimination ability in near duplicate image retrieval tasks, could add robustness to a model in classification tasks.

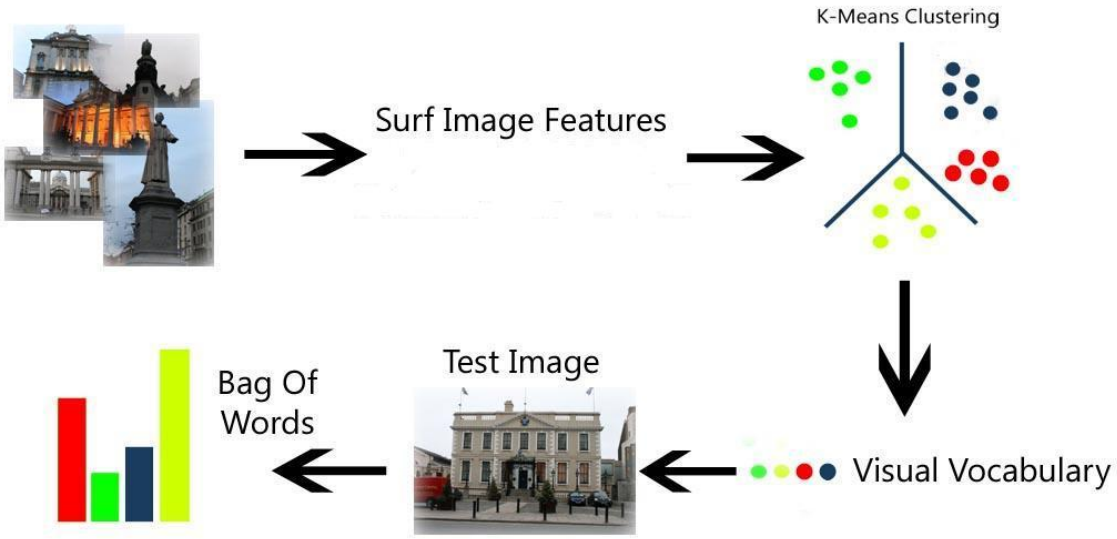

Figure 5.6: An illustration of the process of calculating a visual word histogram from a collection of sample images.

Assignment of Visual Word Features

Traditionally in the VBOW model, image features are assigned to their closest neighbour in the vocabulary, and only their closest neighbour. This assignment process is referred to as 'hard assignment' and can be formally defined as:

$$Hist(i) = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1, & \text{if } i = argmin(Dist(v, p_j) \\ 0, & \text{otherwise} \end{cases}$$

where n is the number of interest points extracted from an image, p_j represents an interest point j extracted from an image, and $Dist(v, p_j)$ represents the distance between a vocabulary word i and an interest point j. This hard assignment model has many disadvantages, in that for each input feature, its similarity is only considered for the closest neighbour in feature space. This model disregards all other features that could be also quite similar, and some of these might only be marginally further away in feature space than the nearest neighbour. Clearly this approach is not ideal as relevant information that can aid discrimination of each feature is simply disregarded.

One method to address this shortcoming is to utilise an approach to feature assignment based on the similarity of features to each vocabulary word, called soft assignment, illustrated in Figure 5.7. This is where each input can be assigned to k bins in a histogram, where k represents its nearest neighbours in feature space. It is desirable to set the amount of the value assigned to each neighbour's bin to be directly proportional to how close the input feature is to each of its neighbours.

Several functions have been proposed and evaluated to calculate the proportional value to be assigned to the k nearest neighbour bins [Viitaniemi and Laaksonen, 2009] such as:

- The inverse Euclidean distance: $||v_i - v||^{-1}$
- The squared inverse Euclidean distance: $||v_i - v||^{-2}$

- The exponential of Euclidean distance: $exp(-\alpha_{\exp}\frac{||v_i-v||}{d_0}$
- The Gaussian function: $\exp(-\alpha_g\frac{||v_i-v||^2}{d_0^2})$

where d_0 is defined as the average distance between two neighbouring vocabulary features. Viitaniemi and Laaksonen [Viitaniemi and Laaksonen, 2009] analysed the effectiveness of this assignment functions in an object classification task using the PASCAL(VOC) 2007 collection of images. They showed that the soft assignment of visual word features to histograms outperformed hard assignment in every category of experiments that they carried out. In similar experiments, Van Gemert et al. [van Gemert et al., 2010] also showed that soft assignment significantly outperformed hard assignment for object classification. Also, what is evident from these experiments is that there are minimal performance differences between the four assignment functions that they tested.

Based on the work of [Viitaniemi and Laaksonen, 2009], among others, a soft assignment visual word approach was implemented and analysed for the purposes of landmark classification. In this work, a ranking based soft histogram assignment function is used. This is mainly due to the fact that this ranking function is quite quick to compute and performs favourably along with some of the more complex assignment functions. This ranking based feature essentially calculates the proportionality of the score assigned to the k nearest neighbour bins, by using the position of each bin in a ranked list (of length k) of nearest neighbours to designate the score. This ranking based function can be formally defined as:

$$Hist(k)+=\frac{1}{2^i-1}$$

where i is the position of the bin k in the ranked list of nearest neighbours.

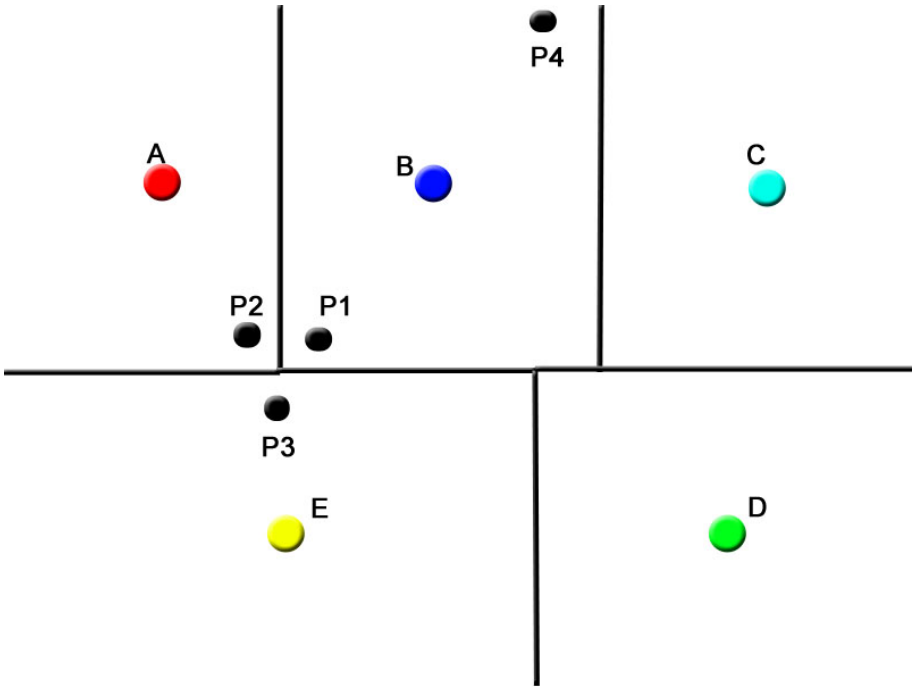

Figure 5.7: An illustration outlining the advantages of the soft assignment of visual word features. This diagram presents a hypothetical partition of a visual word vocabulary containing 5 visual word features A-E, and four feature points to be assigned to a visual word cluster center, P1-P4. It can be clearly seen that points P1, P2 and P3 are quite close together in feature space, however, using a hard assignment approach would not take into account the similarity between these features and they would never be matched. P1 would only be associated with the visual word B, while points P2 and P3 would only be associated with the visual words A and E respectively. Using hard assignment, the only point to be matched to P1 would in fact be P4, even though P2 and P3 are closer in feature space. Using soft assignment, the points P1, P2 and P3 would be assigned to each the visual words A, B and E, albeit, with different weights which are calculated based on the distance from the visual words. This would allow these features to be matched as they are closer in feature space.

Spatial Pyramid Features

VBOW features have been used in the computer vision community for a wide variety of image classification tasks. These features provide an indication of the presence of image patches that are visually similar to their nearest neighbours within the BOW codebook. However, VBOW histograms are not ordered, and more importantly there is no spatial information provided about how the features relate to each other geometrically. A technique is desired that will retain some

information about the spatial layout of visual words. One approach that has been suggested to address the shortcomings of standard visual word features is the use of spatial pyramids .

A spatial pyramid is a feature structure that augments the traditional visual BOW histogram with additional geometrical information. The spatial pyramid is a tree-like structure where the image is partitioned into smaller sub-images using block based segmentation (described in section 4.3.6) at each level of the tree. Each of these sub-images represent finer and finer spatial regions as one traverses down the the tree. The feature is the concatenated histograms from each spatial region at each level of the tree. This is illustrated in Figure 5.8.

While the use of spatial information can be advantageous in many situations, there are also many drawbacks with incorporating spatial information into the BOW model. The main disadvantage is the memory overheads that accompany a much larger feature vector. If the spatial pyramid has 3 levels with a division of 1 $\times$ (2 $\times$ 2) $\times$ (4 $\times$ 4), this will create a feature vector with a length of $k \times 1 \times (2 \times 2) \times (4 \times 4)$, where k is the size of the vocabulary being used. The extra complexity involved with feature vectors of this length can also be immense. Although the process of training SVM models happens offline, the additional complexity involved with spatial pyramids, particularly when using a large value of k, can mean that the training process of each model can take hours, or even days in the case of large models. While offline processing complexity is not as important as the classification or matching processes in an image retrieval/matching framework, the extra processing time involved with spatial pyramids must be considered in any large scale SVM model based system.

Another potential issue is that candidate images could be misclassified if they do not adhere to the spatial constraints of a model. Additionally, the spatial pyramids used in this work (similarly to the other block based segmentation feature used in this section - MPEG7 Edge Histogram), are rotation invariant.

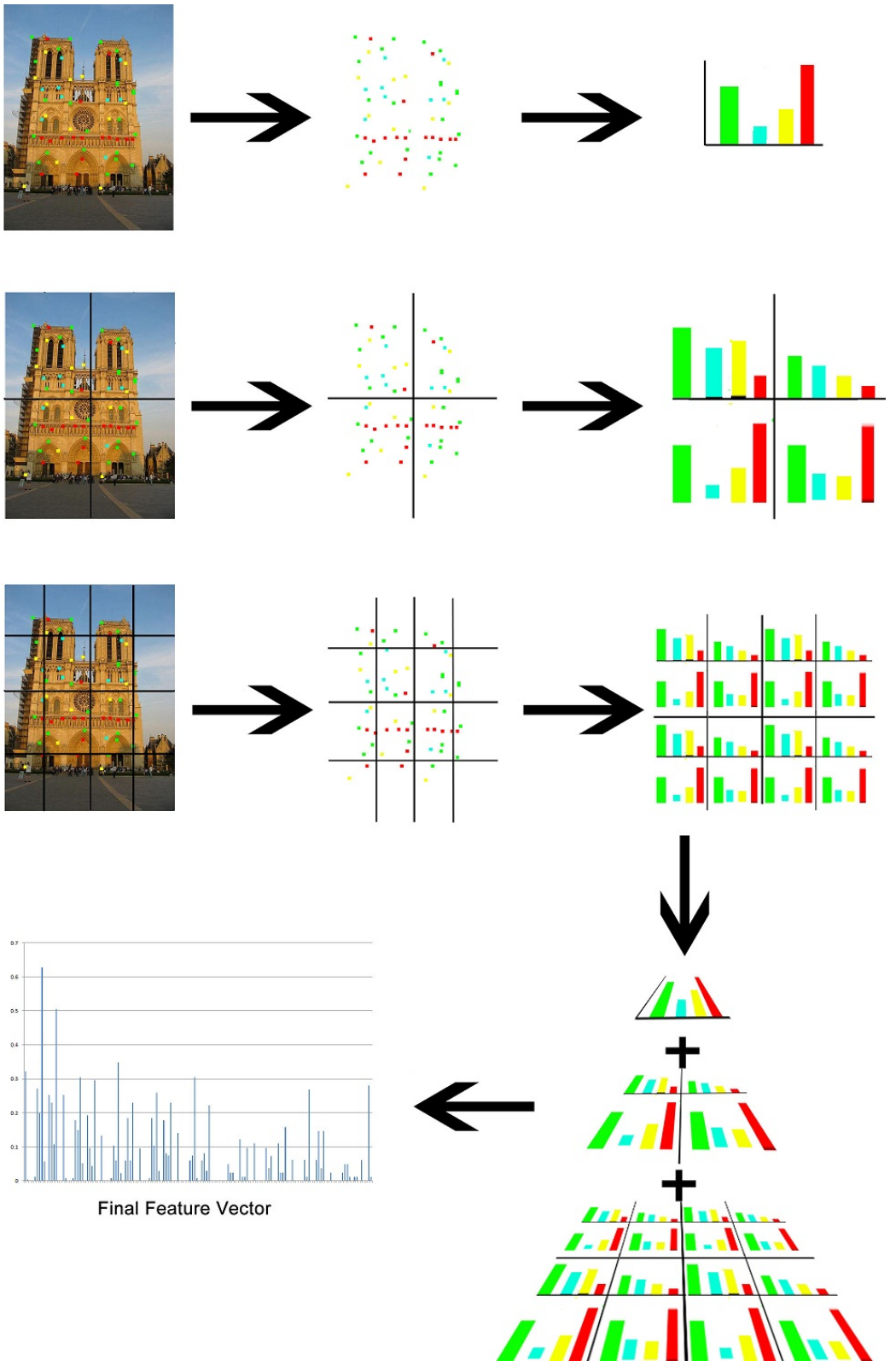

Figure 5.8: An illustration of the process of calculating a spatial pyramid feature vector with a 3 levels and a visual word vocabulary size of 4.

This addition of a weak geometric consistency check could potentially effect the robustness of a classifier.

5.3 Landmark Classification with Supervised Clustered Imagery

The evaluation section at the end of this chapter measures the performance of the machine learning framework (described in section 5.1), however, the evaluation only measures the final outputs of the system. These outputs will be the results of a SURF correspondence process after being processed through the machine learning algorithm. While the final outputs provide a good indication of the accuracy the machine learning methods, there is no specific evaluation to ascertain the exact classification accuracy from the SVMs. In this section, the aim is to evaluate a set of classification models to ascertain a precise measure of how accurately this approach can classify an image of a landmark.

In this work, the aim is to classify between a potentially very high number of instances of a small number of object classes. This task brings with it significant challenges. It is not known in advance how well SVM models would be able to discriminate between these different instances of landmarks which are members of the same class. In this section, experiments were carried out to evaluate how well a machine learning algorithm can distinguish between groups of landmark images. It was infeasible to manually cluster the entire image corpus into clusters, therefore, a manual clustering process was carried out on a different dataset from the main Paris corpus. As this manual clustering was carried out by a human annotator, it is expected that it would produce optimal results. Additionally, in this section, it was desired to discern the effects that affine variation might have on the classification of landmarks using machine learning techniques. This will

help to select optimal parameters to use in the clustering phase that is necessary to create the training data for the machine learning framework.

A new dataset of images was collected for this purpose. This dataset consisted of a large collection of some of the most commonly photographed landmarks in the world (see list below). This dataset was created as it would not be feasible to manually trawl through the Paris corpus and to cluster images manually into viewpoint clusters, as the dataset is very large. Additionally, it is intended to analyse the effects of affine transformations between instances of images within the same class, and this requires a large amount of data for each landmark, and it is desired to determine how this will perform for a range of visually different landmarks.

One of the big advantages of creating a new dataset consisting solely of very well known landmarks is that they can be searched for efficiently online, for example by using the Flickr API using the landmark name as the query text. The hit-ratio of candidate images within this returned set could be quite high, as images returned have a high probability of being relevant to the query. Additionally the advantage of using a dataset consisting of commonly photographed landmarks is that there are a large number of images for each landmark available from Flickr.

A collection of images was collected representing 42 of the most commonly photographed landmarks in the world. For each of these landmarks, 100 near-identical training images was gathered, along with 10 testing images containing the landmark from the same viewpoint. Each of the training and testing images were photographed at similar viewpoints (in the human observers estimation, all were photographed within an affine variation of 45 degrees) These landmarks were selected specifically to represent a diverse range of different types of landmarks, such as churches, bridges, buildings and statues. The 42 landmarks

selected were:

Arc De Triomphe	Astroclock (Prague)	Atominum (Brussels)
Ayers Rock	St Basil's Cathedral	Brandenburg Gate
Buckingham Palace	Christ the Redeemer	Colloseum
Dome, Reichstag Building	Golden Gate Bridge	Golden Temple (Kyoto)
Hagia Sofia	Helsinki Cathedral	Il Duomo (Florence)
Il Duomo (indoor)	Kiyumizu (Kyoto)	Louvre Pyramid
Lincoln Monument	Machu Picchu	Notre Dame Cathedral
Neuschwanstein Castle	Osaka Castle	Parthenon (Athens)
Petronas Towers	Leaning Tower (Pisa)	Ponte Vecchio
Prague Castle	Reichstag Building	Rialto Bridge (front)
Rialto Bridge (side)	Sacre Coeur Cathedral	Santa Maria Novella
Arc De Triomphe	Astroclock (Prague)	Atominum (Brussels)
Statue of David	Statue of David (Outdoor)	Statue of Liberty
Stonehenge	Taj Mahal	Tower Bridge
Trevi Fountain	Westminster (Palace)	

A binary SVM model was trained for each landmark. VBOW features were evaluated using 4 values for k: 256, 512, 1024 and 2048. Each SVM was trained using a one versus all scheme, which is a training method where features from all the class samples are used as positive inputs, while features from all other classes are used as negative inputs. To evaluate the accuracy of machine models using optimally clustered image data, each test image was processed through its associated SVM model. The results of this evaluation can be seen in Figures 5.9, 5.10, 5.11 and 5.12.

From these results, it can be seen that the majority of landmarks can be classified with a high degree of accuracy. Using the highest performing vocabulary size, all classification models achieved an accuracy of over 70%. This clearly demon-

strates that SVMs can be successfully trained to recognise landmark images from a training collection of visually similar images.

5.3.1 Measuring the Effects of Viewpoint Variation

It is important to know how large variations in viewpoint will effect the classification accuracy of SVMs. This information will allow for optimal parameters to be selected in a clustering stage to create sets of training data. More accurate data should ensure a higher classification accuracy. It is desired to discern when training classification models, whether training sets with smaller numbers of near-identical images outperform larger training sets taken from a large variation of viewpoints. In this section, a subset of the 42 landmarks used above was selected to evaluate the effects that large amounts of affine variation in the training collections might have on the accuracy of a classification model. This subset consisted of 10 landmarks. For each landmark two sets of training data were created. The first set consisted of training images that were photographed from a large variety of viewpoints and had a large affine variation. The second set consisted of a group of training images that were all photographed from a similar viewpoint and had a small amount of affine variance. Two sets of test images were collected for each landmark in a similar manner. One test set had a large amount of affine variance and the other contained images taken from the same viewpoint as the second training collection. For each landmark two SVM models were trained using the training data and the test collections were processed through their associated SVM. The results of this experiment are presented in Figures 5.13 and 5.14. From these results, it is evident that there was a large decrease in classification accuracy when using a training set containing large affine variation over the set with a small affine variance.

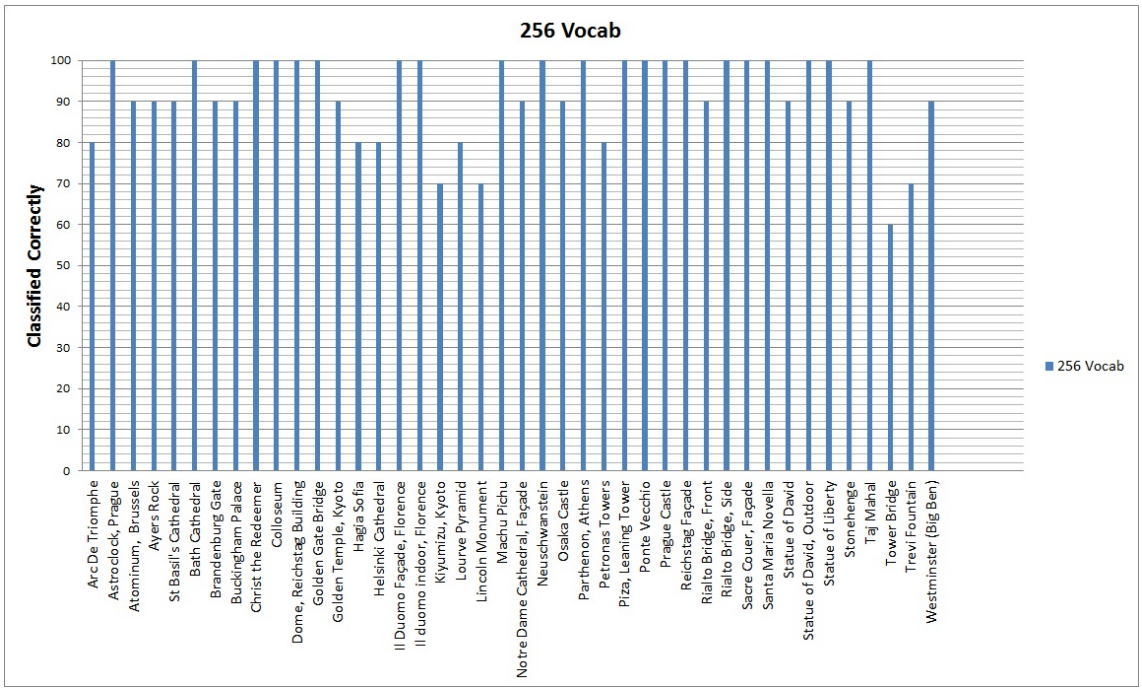

Figure 5.9: **Single Viewpoint Classification.** A diagram illustrating the classification results of the 42 landmarks using a visual vocabulary size of 256.

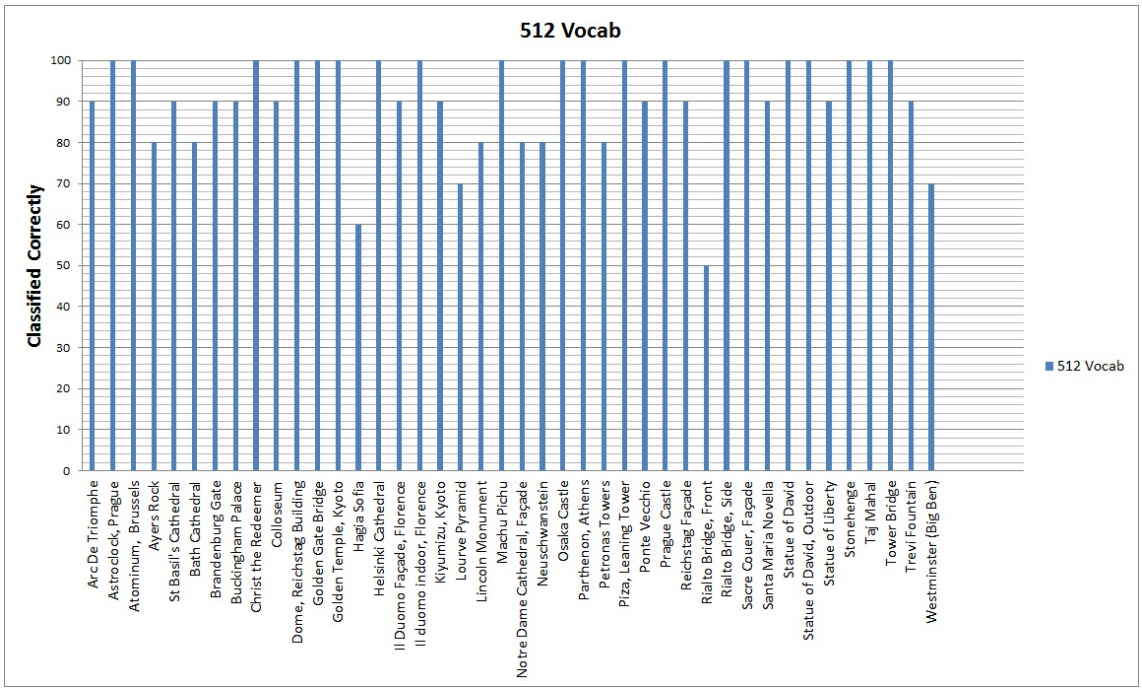

Figure 5.10: **Single Viewpoint Classification.** A diagram illustrating the classification results of the 42 landmarks using a visual vocabulary size of 512.

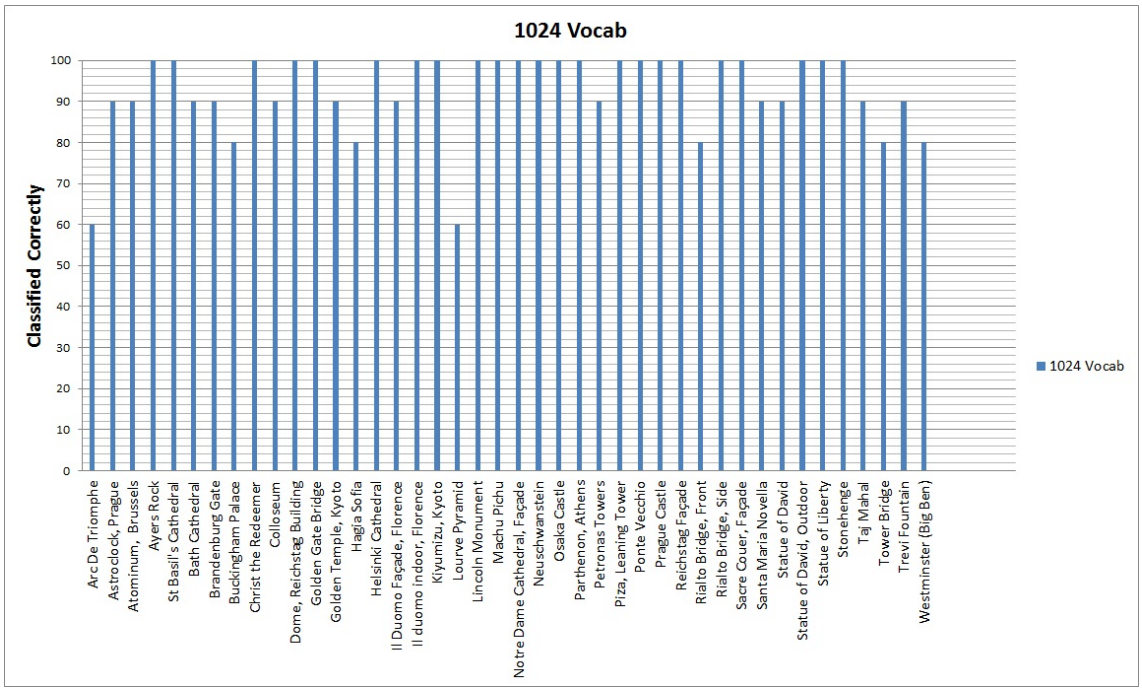

Figure 5.11: **Single Viewpoint Classification.** A diagram illustrating the classification results of the 42 landmarks using a visual vocabulary size of 1024.

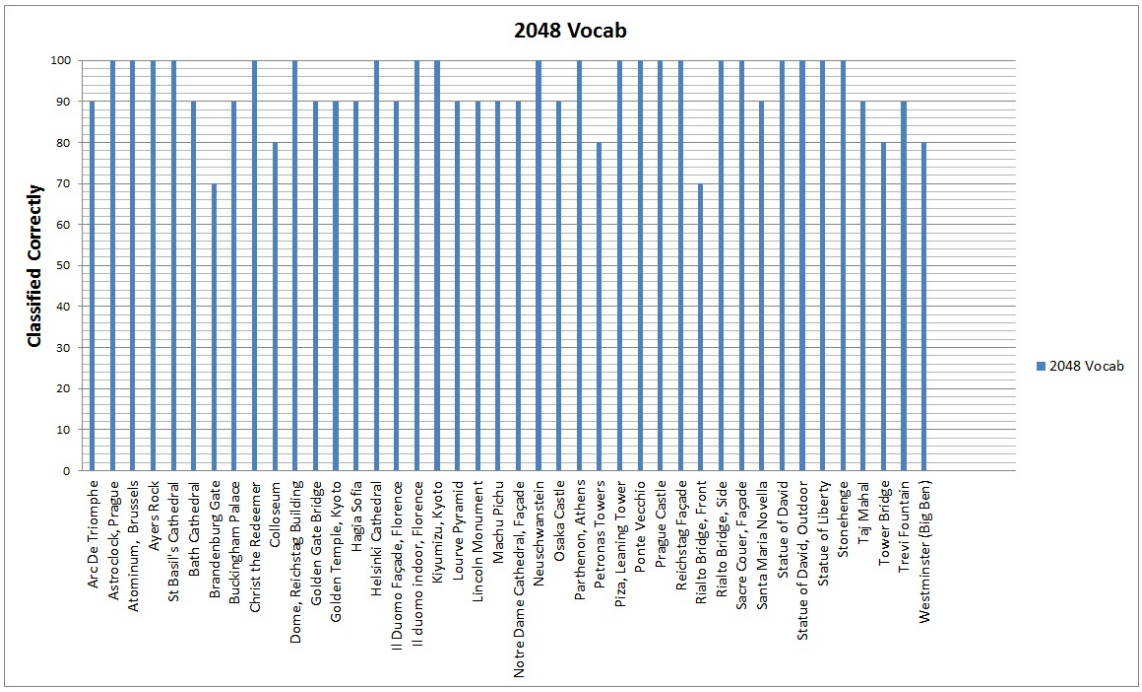

Figure 5.12: **Single Viewpoint Classification.** A diagram illustrating the classification results of the 42 landmarks using a visual vocabulary size of 2048.

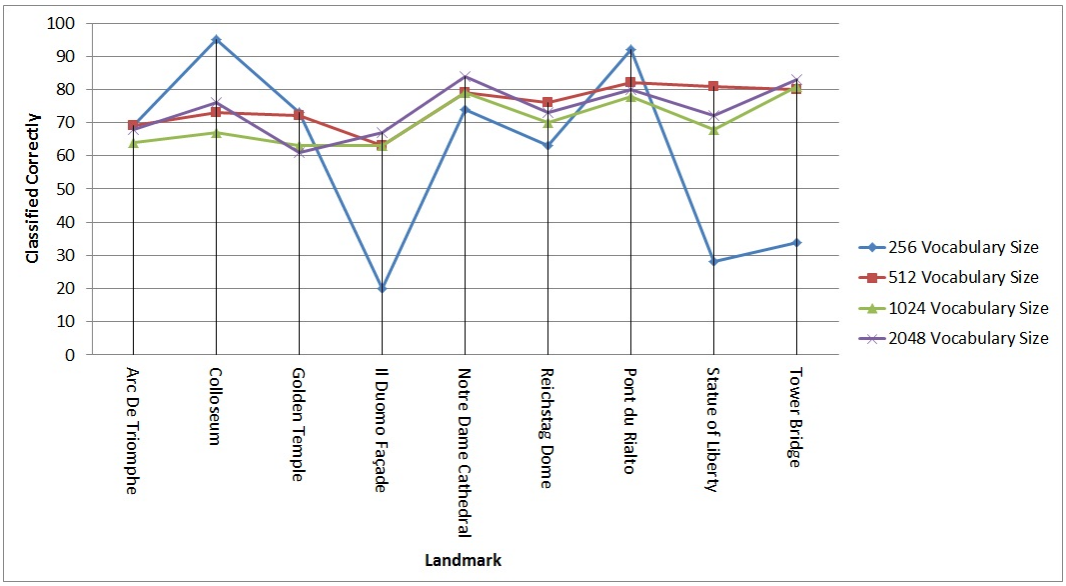

Figure 5.13: An illustration of results from the multi-view classification experiments. The training images in these experiments differed greatly, and contained the associated landmark from a wide variety of angles and viewpoints.

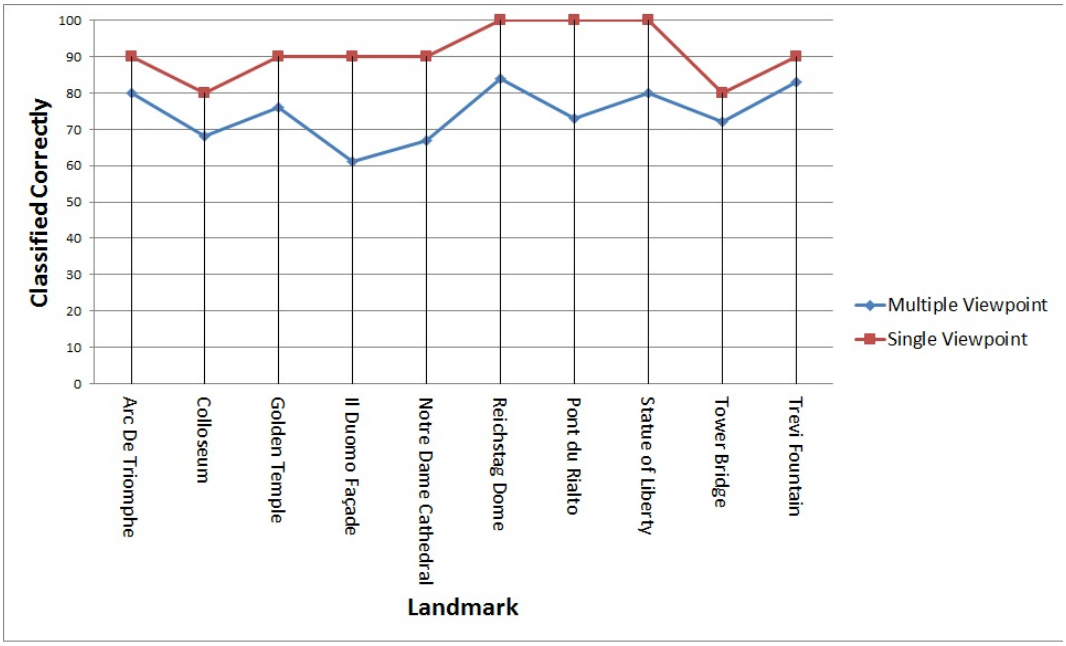

Figure 5.14: A comparison of the percentage of images classified correctly using single viewpoint SVMs versus multiple viewpoint SVMs for 10 landmarks.

5.4 SVM Evaluation

The aim of the framework presented in this thesis is to provide the means to create a memory efficient near real-time landmark classification system. The main focus of evaluating its performance rests with the precision of the framework as opposed to the recall values. While recall is important, particularly in relation to accurate caption and tag selection. In this work, it is viewed as a secondary benefit. It is believed that users of a image classification system would appreciate faster recognition with a small number of accurate results than slower recognition times for a higher number of relevant results, and the framework is designed for this objective.

Based on this belief, the main evaluation objective is to measure the precision of classification results returned by the system. These precision values are created when compared against a benchmark approach, similar to the evaluation approach adopted in Chapter 4.

To create the benchmark, the test collection of 1000 images (described in section 3.2.4) were processed using the benchmark approach described in section 4.5.1. All of the machine learning approaches implemented in this chapter are then evaluated against this benchmark.

The evaluation metrics adopted are:

- Precision (Average) - Precision is defined as the number of relevant images matched divided by the total number of images matched. It is desired to ascertain how accurately each approach performs therefore to calculate the precision metric the average precision value is calculated only using the test images where at least one image was retrieved from the corpus as a match, while all images that had zero matches were disregarded.

- Image Recall - The image recall measure is defined as the normalised percentage of the images within the test set, where at least 3 correct matched

images were returned. Traditional recall metrics measure the percentage of relevant images that are retrieved, however, in this work the aim is not to achieve a high recall, the aim to achieve a high precision across as many query images as possible. The number of images successfully matched is deemed to be more important than the number of relevant images retrieved for each image, therefore, it was believed that a metric measuring the percentage of successfully matched test images was a more relevant in this work.

- Image Recall (Relevant) - For some of the images in the test collection, the benchmark classification process was unable to retrieve a relevant image for them. This could be down to inaccurate geographical data, there was no match for the image with the corpus, or failure of the benchmark approach to recognise them. The image recall (relevant) score is calculated based on the total number of images that at least 3 correct image matches were retrieved for, from only the images within the dataset that had results returned, using the benchmark approach.

- Precision Top 3 - Precision(3) - The Precision(3) ranking calculates the average precision score for only the top 3 ranked images for each test image.

- Precision Top 5 - Precision(5) - The Precision(5) ranking calculates the average precision score for the top 5 ranked images only for each test image.

- Precision Top 10 - Precision(10) - The Precision(10) ranking calculates the average precision score for only the top 10 ranked images for each test image.

- Recall - As described in Chapter 4, recall is a measure of the percentage of the total relevant images within the dataset that were returned for each query image. In this work, recall is not deemed as important as precision,

as it is more important to return a small number of accurate images, rather than a larger number of images with a low precision. Therefore, recall is only considered as a secondary metric, as the main aim of this work is to achieve a high precision score, particularly in the top percentiles of the returned ranked results, i.e. Precision (3). It is expected that the recall score will be quite low for each query image, as during the clustering process a significant proportion of relevant images will be disregarded. Additionally, as these classification models only represent the commonly photographed landmarks within the training corpus, a large number of the test collection will not be classified by this approach.

- Classification Time - As it is envisaged that this framework could be implemented to achieve real-time recognition of landmarks, the required time to process an image is an important attribute to evaluate for each approach. The classification time is defined as the mean number of milliseconds that it required to process all test images. All experiments in this work were carried out on a desktop machine running Windows 7, with an Intel Core 2 Duo, 2.4 GHz chipset and 2 GB of ram. All algorithms were implemented in the Java programming language. Since Java code is converted to bytecode and executed in a runtime environment, it must be noted that these approaches could be further optimized and running times reduced if implemented in a compiled language such as C.

5.4.1 Global Low Level Image Features

Due to the sensitivity of colour features to illumination changes, and the inapplicability of shape based low-level features for the purposes of recognising landmark objects within images, it was decided that texture based features might perform the most reliably out of all types of low-level image features. In this section,

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.434	0.452	0.395	0.452
Image Recall	.356	.349	.352	.349
Image Recall (Relevant)	.429	.421	.425	.421
Precision(3)	0.602	0.619	0.588	0.618
Precision(5)	0.434	0.452	0.412	0.518
Precision(10)	0.434	0.452	.0.403	0.518
Recall	.011	.013	.011	.003
Classification Time (ms)	2575	2665	2691	2701

Table 5.1: Classification results: MPEG7 Edge Histogram

Number of Neighbours	k = 1	k = 5	k = 9	k = 15
Precision(Overall)	0.709	0.693	0.698	0.703
Image Recall	.313	.341	.349	.352
Image Recall (Relevant)	.378	.411	.421	.425
Precision(3)	0.866	0.857	0.862	0.870
Precision(5)	0.832	0.819	0.826	0.827
Precision(10)	0.774	0.761	0.768	0.771
Recall	.074	.070	.068	.070
Classification Time (ms)	2575	2665	2691	2356

Table 5.2: k-NN Classification results: MPEG7 Edge Histogram

experiments were carried out using a texture based feature as an input into classification models. The feature evaluated was the MPEG7 Edge Histogram feature which is described in section 4.3.6.

As expected the low-level feature based approach performed poorly when using SVM models. A large number of images were classified using this approach, however, the precision scores were low. It was found that if combining low-level features with a k-NN classifier, the results were above the average of all evaluated approaches. It is assumed that the nearest neighbours in feature space were quite close visually, and therefore, using a small value for k in the k-NN classifier, allows for these images to be classified correctly.

5.4.2 Visual Bag of Word Histograms

In this section, the machine learning classification system was evaluated using VBOW histograms with hard assignment as the input features. Three values were evaluated for k; 1024, 2048 and 4096. Two machine learning algorithms were evaluated for each of these values, SVMs and K-NN. The results of this evaluation can be seen in Tables 5.5 - 5.10.

Of all of the approaches evaluated, the VBOW hard assignment performed with the highest level of classification precision and additionally has the highest image recall scores. This is illustrated in Figure 5.15. Based on the results in Table 5.7, it can be seen that 399 images of the test collection were successfully classified using this method.

The training data used to create the classification models contained a lot of near duplicate images. This meant that as opposed to solving a scene classification task, effectively by the nature of the clustered data, the problem being solved here was an specific object recognition task, with little visual variation between training images. It is therefore logical that a straightforward measure of the distribution of visually similar image patches would outperform a noisier feature such as that provided by VBOW with soft assignment. Although, it is expected that soft assignment would outperform hard assignment in a scene classification task.

From analysing the results in Tables 5.5 - 5.10 it can be seen that the hard assignment visual word features not only achieve a higher precision(3) score but higher precision scores as the lower ranking matches are measured. This would indicate that the algorithms using these features are selecting the optimal class from the multi-class model on more occasions than other features.

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.688	0.651	0.695	0.685
Image Recall	.363	.333	.405	.395
Image Recall (Relevant)	.438	.402	.489	.477
Precision(3)	0.855	0.849	0.860	0.853
Precision(5)	0.815	0.808	0.824	0.814
Precision(10)	0.757	0.737	0.769	0.757
Recall	.060	.053	.068	.066
Classification Time (ms)	2990	3372	2899	2980

Table 5.3: SVM Classification results: Visual BOW (k = 1024)

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.678	.651	0.688	0.688
Image Recall	.356	.333	.396	.399
Image Recall (Relevant)	.429	.402	.478	.481
Precision(3)	0.841	0.849	0.861	0.868
Precision(5)	0.792	0.808	0.823	0.826
Precision(10)	0.729	0.737	0.763	0.762
Recall	.061	.053	.068	.066
Classification Time (ms)	3102	3542	3029	3212

Table 5.4: SVM Classification results: Visual BOW (k = 2048)

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.709	0.651	0.712	0.696
Image Recall	.397	.333	.399	.398
Image Recall (Relevant)	.479	.402	.481	.480
Precision(3)	0.871	0.849	0.873	0.868
Precision(5)	0.838	0.808	0.841	0.831
Precision(10)	0.776	0.737	.0.777	0.766
Recall	.071	.053	.069	.071
Classification Time (ms)	3470	3934	3508	3656

Table 5.5: SVM Classification results: Visual BOW (k = 4096)

Number of Neighbours	$k = 1$	$k = 5$	$k = 9$	$k = 15$
Precision(Overall)	0.709	0.702	0.698	0.694
Image Recall	.356	.349	.352	.349
Image Recall (Relevant)	.429	.421	.425	.421
Precision(3)	0.866	0.868	0.860	0.860
Precision(5)	0.830	0.834	0.828	0.823
Precision(10)	0.774	0.775	.0.771	0.765
Recall	.074	.079	.076	.081
Classification Time (ms)	2575	2665	2691	2701

Table 5.6: k-NN Classification results: Visual BOW (k = 1024)

Number of Neighbours	$k = 1$	$k = 5$	$k = 9$	$k = 15$
Precision(Overall)	0.718	0.710	0.705	0.705
Image Recall	.434	.331	.359	.368
Image Recall (Relevant)	.429	.421	.433	.444
Precision(3)	0.876	0.871	0.873	0.871
Precision(5)	0.835	0.838	0.838	0.835
Precision(10)	0.775	0.776	0.775	0.773
Recall	.066	.074	.075	.074
Classification Time (ms)	2657	2719	2800	2868

Table 5.7: k-NN Classification results: Visual BOW (k = 2048)

Number of Neighbours	$k = 1$	$k = 5$	$k = 9$	$k = 15$
Precision(Overall)	0.697	0.706	0.707	0.699
Image Recall	.352	.356	.360	.359
Image Recall (Relevant)	.425	.429	.434	.433
Precision(3)	0.860	0.868	0.871	0.870
Precision(5)	0.827	0.840	0.840	0.838
Precision(10)	0.768	0.778	0.776	0.775
Recall	.079	.078	.077	.079
Classification Time (ms)	2781	2939	3001	3078

Table 5.8: k-NN Classification results: Visual BOW (k = 4096)

5.4.3 Visual Bag of Word Histograms with Soft Assignment

In this section, the SVM classification system was evaluated using VBOW histograms with soft assignment as the input features. Three values were evaluated for k: 1024, 2048 and 4096. Two machine learning algorithms were evaluated for each of these values. The results of this evaluation are presented in Tables 5.5 - 5.10.

From the results of this evaluation, it has been found that the soft assignment method does not perform as well as using the standard hard assignment method when creating VBOW histograms. The only soft assignment approach that achieved acceptable precision and image recall scores was when using a vocabulary size of 4096. At smaller sizes, the feature seemed only to be able to separate very dominant clusters in the dataset. When using the values of 1024 and 2048 for k, the system seems to classify a similar number of images as when using the spatial pyramid features described in Tables 5.15 - 5.17. This would suggest that when using classification models trained on noisy features, that the algorithm fails to partition the dataset effectively and only dominant image clusters are recognised correctly.

5.4.4 Spatial Pyramid

In this section the machine learning classification system was evaluated using spatial pyramid features as the inputs to the classification algorithms. Three sizes were evaluated for k: 128, 256 and 512. The pyramid feature evaluated in this work consisted of 3 levels with a block segmentation of $1 \times (2 \times 2) \times (4 \times 4)$. This consisted of feature vector lengths of 2688, 5376 and 10,572. The large size of these

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.694	0.694	0.694	0.694
Image Recall	.235	.229	.233	.233
Image Recall (Relevant)	.283	.276	.281	.281
Precision(3)	0.849	0.855	0.854	854
Precision(5)	0.808	0.812	0.810	810
Precision(10)	0.746	0.746	.0.746	0.746
Recall	.047	.047	.046	.046
Classification Time (ms)	2950	3372	2899	2980

Table 5.9: SVM Classification results (Soft Assignment): Visual BOW (k = 1024)

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.692	0.694	0.694	0.694
Image Recall	.235	.229	.233	.233
Image Recall (Relevant)	.283	.276	.281	.281
Precision(3)	0.848	0.855	0.854	0.854
Precision(5)	0.807	0.812	0.810	0.810
Precision(10)	0.744	0.746	.0.746	0.746
Recall	.047	.047	.046	.046
Classification Time (ms)	3550	3542	3029	3212

Table 5.10: SVM Classification results (Soft Assignment): Visual BOW (k = 2048)

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.697	0.673	0.684	0.682
Image Recall	.369	.301	.357	.355
Image Recall (Relevant)	.445	.363	.431	.428
Precision(3)	0.861	0.855	0.858	0.832
Precision(5)	0.825	0.820	0.821	0.802
Precision(10)	0.766	0.754	0.760	0.744
Recall	.072	.056	.064	.186
Classification Time (ms)	3470	3934	3508	3656

Table 5.11: SVM Classification results (Soft Assignment): Visual BOW (k = 4096)

Number of Neighbours	k = 1	k = 5	k = 9	k = 15
Precision(Overall)	0.687	0.687	0.687	0.687
Image Recall	.244	.244	.244	.244
Image Recall (Relevant)	.294	.294	.294	.294
Precision(3)	0.877	0.877	0.877	0.877
Precision(5)	0.829	0.829	0.829	0.829
Precision(10)	0.754	0.754	.0.754	0.754
Recall	.049	.049	.049	.049
Classification Time (ms)	2203	2129	2127	2126

Table 5.12: k-NN Classification results (Soft Assignment): Visual BOW (k = 1024)

Number of Neighbours	k = 1	k = 5	k = 9	k = 15
Precision(Overall)	0.687	0.687	0.687	0.687
Image Recall	.244	.244	.244	.244
Image Recall (Relevant)	.294	.294	.294	.294
Precision(3)	0.877	0.877	0.877	0.877
Precision(5)	0.829	0.829	0.829	0.829
Precision(10)	0.754	0.754	.0.754	0.754
Recall	.049	.049	.049	.049
Classification Time (ms)	2284	2274	2279	2273

Table 5.13: k-NN Classification results (Soft Assignment): Visual BOW (k = 2048)

Number of Neighbours	k = 1	k = 5	k = 9	k = 15
Precision(Overall)	0.687	0.687	0.687	0.687
Image Recall	.244	.244	.244	.244
Image Recall (Relevant)	.294	.294	.294	.294
Precision(3)	0.877	0.877	0.877	0.877
Precision(5)	0.829	0.829	0.829	0.829
Precision(10)	0.754	0.754	.0.754	0.754
Recall	.049	.049	.049	.049
Classification Time (ms)	2566	2544	2551	2536

Table 5.14: k-NN Classification results (Soft Assignment): Visual BOW (k = 4096)

feature vector lengths meant that it required significant processing time to train a set of models using the SVM algorithm. For example, to train a set of models using the RBF kernel with a vocabulary size of 512 required over 220 hours of computing time.

The results in Tables 5.15 - 5.20 suggest that the spatial pyramid features are not suitable to this type of classification. The features are too sparse to be suitable for training machine learning models. A large proportion of the feature vector will remain empty, particularly in the finer segmented regions. A smaller number of interest point features will be detected the smaller the size of the region being analysed. Additionally, many regions within an image might not be very salient and will produce very few interest point features. This is less of an issue when calculating a global based feature such as a VBOW, as it is expected there will be a high distribution of visual word features across an entire image.

The Spatial pyramid features all performed identically for each evaluated classification algorithm, regardless of vocabulary size or classification parameters. Results from Tables 5.15 - 5.17 would suggest that for 226 test images, the correct image class is quite dominant in the feature space of the nearest SVM model. Classification using this feature only works well with these images and performs very poorly for all other classes that are less dominant. Using the feature, the machine learning algorithm is unable to separate the data accurately. Tables 5.18 - 5.20 would also support this theory, as similar results are recorded. Using the k-NN classifier, it would appear that for 244 test images, the correct image class is quite dominant in the feature space of the nearest k-NN model.

Overall, the spatial pyramid features performed poorly. A high precision was recorded for images that were classified using the feature, however, only a small number of images were recognised, most likely the test images that would be expected to be classified correctly due to large representation in the training sets.

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.704	0.704	0.704	0.704
Image Recall	.226	.226	.226	.226
Image Recall (Relevant)	.272	.272	.272	.272
Precision(3)	0.858	0.858	0.858	0.858
Precision(5)	0.814	0.814	0.814	0.814
Precision(10)	0.750	0.750	.0.750	0.750
Recall	.047	.047	.047	.047
Classification Time (ms)	2118	2292	2348	2408

Table 5.15: SVM Classification results : Spatial Pyramid (k = 128)

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.704	0.704	0.704	0.704
Image Recall	.226	.226	.226	.226
Image Recall (Relevant)	.272	.272	.272	.272
Precision(3)	0.858	0.858	0.858	0.858
Precision(5)	0.814	0.814	0.814	0.814
Precision(10)	0.750	0.750	.0.750	0.750
Recall	.047	.047	.047	.047
Classification Time (ms)	2588	2584	2589	2583

Table 5.16: SVM Classification results: Spatial Pyramid (k = 256)

SVM Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Precision(Overall)	0.704	0.704	0.704	0.704
Image Recall	.226	.226	.226	.226
Image Recall (Relevant)	.272	.272	.272	.272
Precision(3)	0.858	0.858	0.858	0.858
Precision(5)	0.814	0.814	0.814	0.814
Precision(10)	0.750	0.750	.0.750	0.750
Recall	.047	.047	.047	.047
Classification Time (ms)	3213	3202	3201	3185

Table 5.17: SVM Classification results : Spatial Pyramid (k = 512)

Number of Neighbours	$k = 1$	$k = 5$	$k = 9$	$k = 15$
Precision(Overall)	0.687	0.687	0.687	0.687
Image Recall	.244	.244	.244	.244
Image Recall (Relevant)	.294	.294	.294	.294
Precision(3)	0.877	0.877	0.877	0.877
Precision(5)	0.829	0.829	0.829	0.829
Precision(10)	0.750	0.750	.0.750	0.750
Recall	.049	.049	.049	.049
Classification Time (ms)	2409	2109	2173	2172

Table 5.18: k-NN Classification results : Spatial Pyramid (k = 128)

Number of Neighbours	$k = 1$	$k = 5$	$k = 9$	$k = 15$
Precision(Overall)	0.687	0.687	0.687	0.687
Image Recall	.244	.244	.244	.244
Image Recall (Relevant)	.294	.294	.294	.294
Precision(3)	0.877	0.877	0.877	0.877
Precision(5)	0.829	0.829	0.829	0.829
Precision(10)	0.750	0.750	.0.750	0.750
Recall	.049	.049	.049	.049
Classification Time (ms)	2750	2763	2749	2746

Table 5.19: k-NN Classification results : Spatial Pyramid (k = 256)

Number of Neighbours	$k = 1$	$k = 5$	$k = 9$	$k = 15$
Precision(Overall)	0.687	0.687	0.687	0.687
Image Recall	.244	.244	.244	.244
Image Recall (Relevant)	.294	.294	.294	.294
Precision(3)	0.877	0.877	0.877	0.877
Precision(5)	0.829	0.829	0.829	0.829
Precision(10)	0.750	0.750	.0.750	0.750
Recall	.049	.049	.049	.049
Classification Time (ms)	3213	3202	3201	3185

Table 5.20: k-NN Classification results : Spatial Pyramid (k = 512)

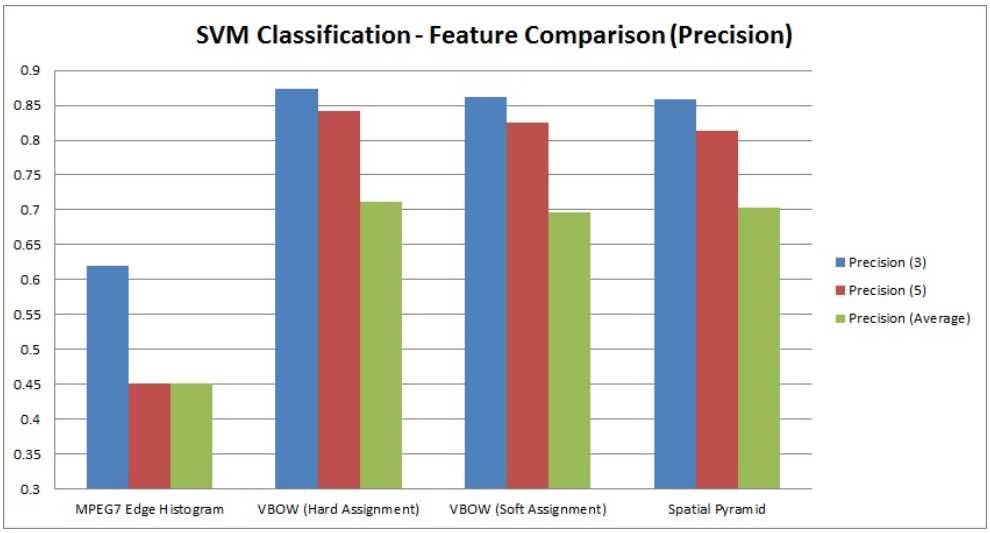

Figure 5.15: An outline of the precision scores for the top performing classification run for each evaluated image feature.

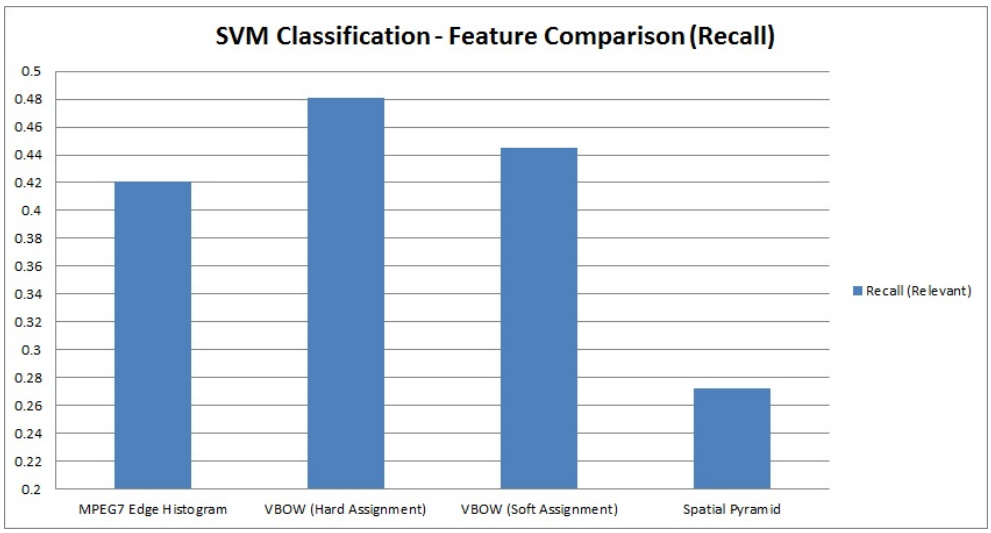

Figure 5.16: An outline of the image recall (relevant) scores for the top performing classification run for each evaluated image feature.

5.5 Conclusion

In this Chapter, a framework was implemented based on spatially organised machine learning classification models. From the evaluation section it is this chapter, it is evident that by structuring models in this manner it is possible to classify query images containing commonly photographed landmarks with a high degree of precision. This method of organising classification models

based on geographical location improves upon previous techniques suggested in the literature [Li et al., 2009b]. Quantising the number of classes used to train each multi-class model, can ensure that classification accuracy will remain high provided the classes can be differentiated using visual features.

Two separate machine learning algorithms were evaluated, SVMs and k-NN. As can be seen from Figures 5.16 and 5.17, there was a marginal difference in the performance between the two algorithms. It would seem that the most important parameter is the actual input feature. Overall, the best performing algorithm was the SVM, however, for many of the classification runs, the k-NN output a higher precision score and had the advantage of a lower processing time. The k-NN classifier with relatively small values of k was chosen to for this evaluation because, by the nature of the clustering process used (described in Chapter 4), the training data for each classification model, contained numerous near-duplicate images. From this observation, it is logical to assume that the training features from the near duplicate images would be located close to each other in feature space. It is expected that, as the visual variation in cluster images becomes larger and the numbers of non near-duplicate images within a cluster increases, the performance of the k-NN would decrease and the SVM would provide a better generalisation performance. From the results of the evaluation it would seem that the SVM decision function is able to perform better with the less dominant image clusters and thus successfully recognises a larger number of test images.

It would seem that using the optimal parameters, just under 50% (48.9%) of all test images can be classified in real-time with a high level of accuracy using the machine learning based classification approach. Given the high precision scores alongside the high image recall, this is deemed to be a good result. The recognition process can be executed in real-time and this approach represents a dramatic decrease (by a factor of over 100) in the time required to classify a test

image, over the brute force interest point matching as utilised in the benchmark approach.

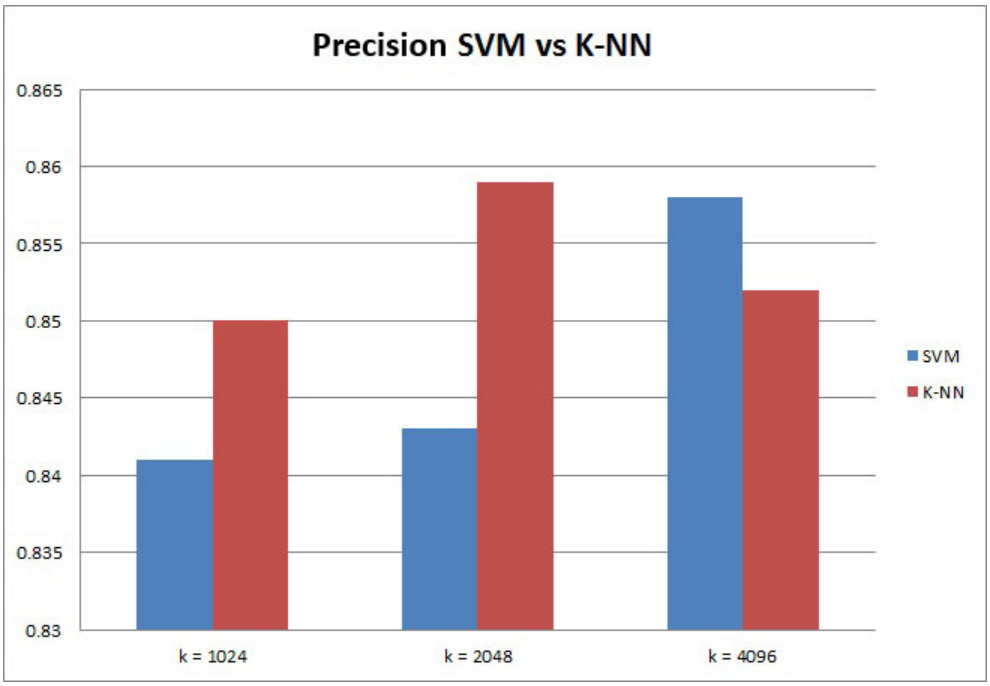

Figure 5.17: An outline of the precision scores for the top 3 ranking results between two different machine learning algorithms (SVM and k-NN) using the optimal input features (VBOW - Hard Assignment). When the VBOW is assigned smaller values for k where k in this case refers to the size of the visual vocabulary, the k-NN will outperform the SVM method.

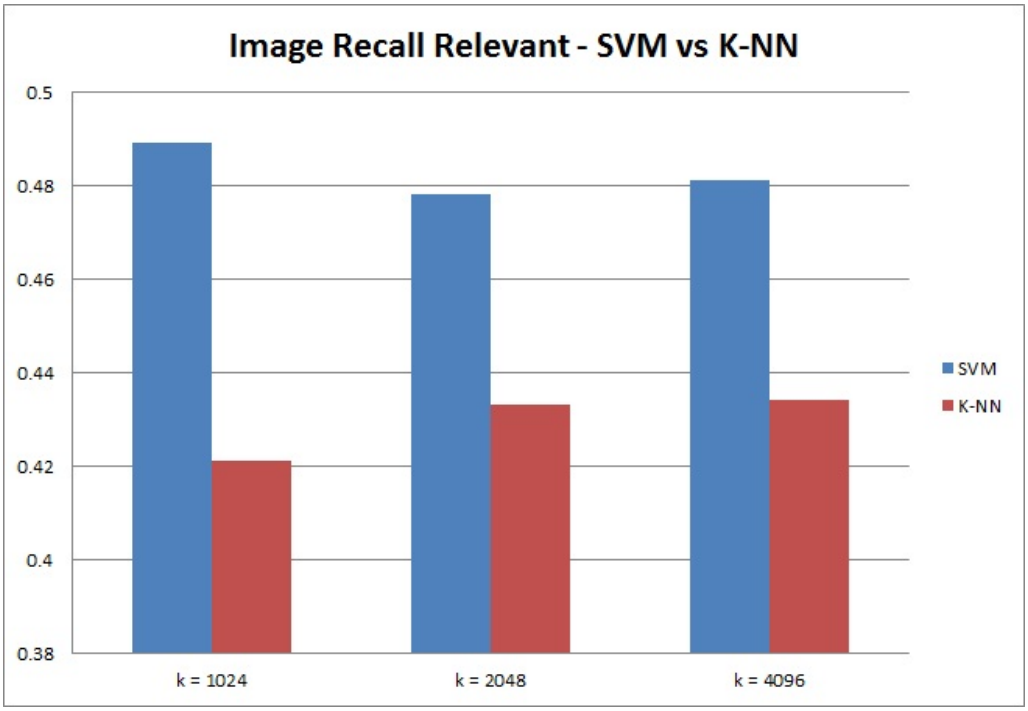

Figure 5.18: An outline of the image recall(relevant) scores between two different machine learning algorithms (SVM and k-NN) using the optimal input features (VBOW - Hard Assignment).

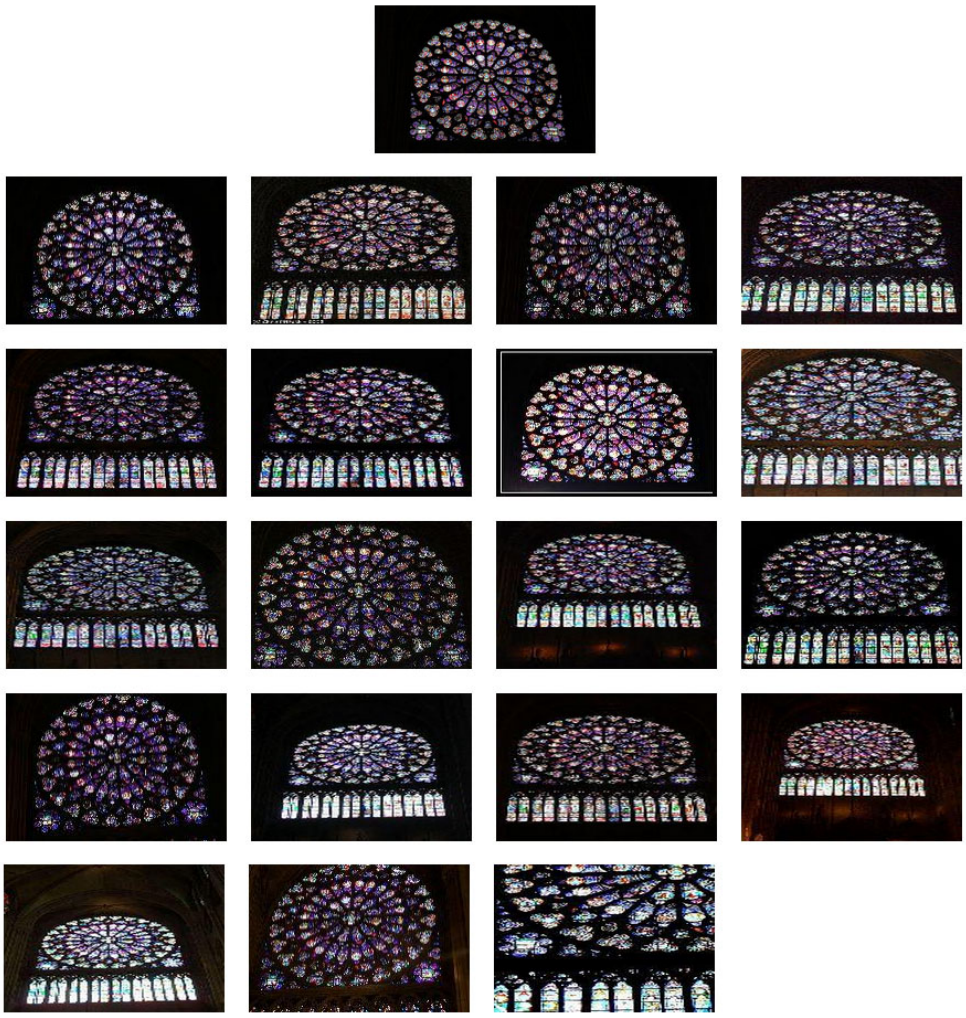

Figure 5.19: An example of the set of retrieved images that were matched with a random test image using the top performing approach (SVM, VBOW (k = 4096)). The test image is displayed at the top and the images are ranked from left to right , top to bottom in terms of image similarity.

Chapter 6

Hybrid Approaches to Landmark Classification

6.1 Introduction

In this chapter, two approaches are proposed to solve the problem of classifying landmark images with a low frequency within the corpus. These images include landmarks that are not commonly photographed, and images of a landmark taken from a non-popular viewpoint. The chapter begins with a description of the first approach evaluated. This approach is based on the use of scene classification models to reduce the search space as part of a nearest neighbour search problem. The second approach is then introduced which is based upon the use of a tree structure to index local image features. The best performing of these two approaches is then combined with the machine learning classification method described in Chapter 5, to form a hybrid system. An evaluation of all 3 approaches is then presented. In this evaluation, these approaches are compared against a current state of the art technique for landmark recognition.

Support Vector Machines require a set of positive and negative examples in order to train a classification model. Although the number of examples is not

explicitly known in advance for a problem, too small a number will produce inaccurate prediction results and lead to high numbers of false positives. There is significant overlap in the number of visually similar images for a commonly photographed view of a popular landmark, there is a need, however, to address the issue of how to classify viewpoints of landmarks that are not commonly captured.

Several techniques developed by the computer vision community for matching visually similar images in large data collections, have been shown to work quite accurately [Sivic and Zisserman, 2006], however, many of the commonly used image matching algorithms cannot be used with very large data sets using standard desktop computers, due to memory constraints. Image feature vectors can be quite large and, in the case of interest points can require more memory to store and process than the actual image itself. Several algorithms used for efficient scene and object matching, such as vocabulary trees, require that the data structure containing the image features be loaded into heap memory before the matching process can be executed. This provides a constraint on the number of training features that will fit into the heap memory. In this section, an image matching approach is used that allows for additional images to be added to a training corpus, while retaining a static memory requirement.

The aim of this chapter is to find the nearest neighbour of a test image within the corpus that could not be classified, or was perhaps mis-classified by the SVM classification approach (presented in Chapter 5). The nearest neighbour is represented in some feature space using a distance metric. In this case the metric used is the L2 norm in Euclidean space.

This may sound like a trivial task that could be computed in linear time i.e. O(n) where n is the number of images within a training corpus. When there are large values for n, however (where a large value is defined as thousands, or perhaps even millions), this exact nearest neighbour approach could not be

achieved in real time. For extremely large values of n, the task would soon become intractable using today's technology.

This issue is particularly important when the distance measure concerns correspondences between sets of interest point descriptors, primarily because hundreds and possibly thousands of local image features can represent a single image, depending on its level of saliency. Therefore, to calculate a distance measurement between a single pair of images could potentially require millions of feature vector comparisons.

The search space in this work is too large to allow for exact brute force nearest neighbour matching of interest point features in real time, an approach is required that will find an approximate nearest neighbour, or that will reduce the size of the search space through the removal of non-relevant matches using cheap distance measures. Two of these proposed approaches to address this problem were analysed and evaluated: Hierarchical Classification using Scene Classification Models and Hierarchical Vocabulary Trees.

6.2 Hierarchical Classification

As was discussed in section 6.1, when matching a test image using point to point approaches based on SURF descriptors, it is not feasible to compare a test image against all images in the corpora, as it would require a significant amount of computing resources. Additionally, the accuracy of the interest point matching will decrease as the size of the corpus increases. It is desirable to first prune the search space using less expensive methods, disregarding non-relevant corpus images. One such method, is to disregard non-relevant images based on spatial locations extracted from GPS information or manually created geo-tags. Although spatial based methods will successfully filter out large numbers of non-relevant

images, issues still exist in areas where there is a high concentration of commonly photographed landmarks.

Many large urban areas, such as the centre of Paris, contain well known landmarks in close proximity to one another. This can cause problems even with spatial filtering techniques as large numbers of images would remain after filtering. This issue is even more prevalent with the dataset used in this work, as the majority of geographical information is set manually by human observers and therefore relatively wide spatial radii (250 meters) need to be used to ensure an acceptable balance between relevant and non-relevant images being retrieved, as described in Chapter 3. It is necessary to employ additional techniques to further reduce the query space before expensive interest point matching methods are processed.

In this section, an approach is proposed to help solve this problem. This approach is based on structuring the search space according to the class or type of landmark that is represented in each image. In a hierarchical image matching scheme, traditionally the aim is to treat this search space reduction problem as an image similarity process, where a corpus image is disregarded if a similarity measure between it, and a query image is above a threshold. This similarity measure ideally can be calculated quickly and efficiently. In this section, it is proposed to analyse the effectiveness of semantic scene classification models, which effectively treats this search space pruning process as a scene classification task. If a corpus can be structured into small spatially organised collections of images that are semantically grouped into different landmark categories, it is proposed that this will allow for efficient pruning of the corpus when matching a query image. In this section, an evaluation is carried out to test this proposal.

6.3 Low-Level Semantic Classifiers and Concept Detection

Understanding the content of a scene depicted within an image is one of the core goals of computer vision. The aim is to convert the pixel data contained within an image into one or more high level semantic descriptions of the scene or event that is displayed in an image. A high level semantic description could be described as a detailed and meaningful representation of the content of an image, which would be relevant to a human observer (or perhaps a description that could be converted by a computer so that it would be relevant to a human observer). High-level semantic image classification is still a very open problem in the computer vision community, particularly in unconstrained environments. In recent years however, much progress has been made in image classification at a lower semantic level, such as the ability to classify images into different categories of scenes.

Following the paradigm of the machine learning approach described in Chapter 5, it was hypothesised that it would be possible to classify an image of a landmark into one of a finite number of visually distinct categories. The motivation behind this is that if accurate semantically relevant groupings of corpus images could be achieved, it would be possible to reduce the size of the search space dramatically in a landmark recognition task, based on these groupings. In this work, 8 different classes of landmarks were chosen that had a high representation within the corpus, and could be suitable for classification using machine learning approaches. These were:

- **Artwork** The artwork class is defined as images (that contain a painting or drawing) taken inside an art gallery or museum. From an informal empirical study of the Paris corpus, it is evident that many Flickr users

commonly photograph paintings, and several well known pieces of art could be considered landmarks.

- **Bridge** Another very commonly photographed landmark is a bridge. Many iconic bridges span the river Seine and many of the canals that flow through the Parisian region, and due to their unique visual appearance and photogenicity, large numbers appear in the training corpus. In this work, a bridge is defined as a man-made object that spans across a body of water, a road or a railway track.

- **Building Facade** A building facade is a category containing the main facade of a large building. If there is no notable facade, for example in the case of an office block or a skyscraper, the facade is considered to be any side of the building.

- **Fountain** A fountain is defined as a man made object that sprays or pours water either into the air or into a man made reservoir. Although originally used for human water consumption purposes, today fountains are mainly used for ornamental purposes.

- **Monument** The category of monument is quite nebulous and can refer to a large number of objects. In this work, a monument is considered a man-made structure that does not have a use as a dwelling place (such as a building) and does not contain a large statue or sculpture. Some examples of monuments in the image corpus are; the 'Eiffel Tower' and the 'Arc de Triomphe'.

- **Church:** A Church is defined as a place where a Christian might practice their religion, such as a church, cathedral or a chapel. This category is concerned solely with images that were taken outside of the structure.

- **Church(Indoor):** The church indoor is defined as an image that was photographed inside a Christian place of worship. These commonly include close up images of stained glass windows, church ornaments and altars.

- **Statue** A statue is defined as a sculpture that usually represents a person or historical event. Additionally, a sculpture within an art gallery or museum will fall into this category.

- **Other** Any landmark that does not fall into one of the above categories is defined in this class.

Although there is a large amount of variance in intra class visual similarity within each of these categories, many different landmarks within a class share some basic characteristics. Take, for example, the class 'Church', which includes churches and cathedrals, among others. In many cases, a human observer could quickly recognise a church as being a church irrespective of the size of the landmark or the architecture style, as illustrated in Figure 6.1. Whether a church was built in the Gothic style, such as the famous Notre Dame Cathedral, or in a more modern style such as the Sagrada Familia in Barcelona, many humans could identify from visual recognition that these structures are places of worship. This recognition could be based on knowledge obtained in their lifetime using the visual style of other visually similar places of worship, which could be considered analogous to supervised learning. It is logical, therefore, to assume that these two structures share enough characteristics visually, for a human observer to predict the category of both structures without heterogeneous knowledge. It is based on this premise, that a suite of classification models was implemented with the aim of grouping landmarks into a finite set of categories.

The ability to quickly classify a landmark category from an image would prove useful in selecting relevant candidates for further analysis from a large corpus in an image matching approach. Effectively this approach is adopting a

scene classification approach to pruning the search space. By using this approach and training a suite of k classification models representing landmark categories, the nearest neighbour search problem would remain linear. It would however, be reduced to a complexity of $O(n/k)$. Combining a suite of classifiers with geographical based pruning can further reduce the complexity of the search algorithm to $O(n/(k \times i))$, where i represents the number of spatial divisions within a corpus. With large values of k and i, a corpus can be quickly reduced to a small subset of relevant images, within which a linear search may be carried out for the nearest neighbour. It must be noted, however, that this is an optimal complexity measurement which assumes a uniform geographical distribution of images within a region, which is not the case in this work. The proposed system based on this approach is illustrated in Figure 6.2.

6.3.1 Visual Semantic Classifier

To create a hierarchical classification system based on semantic classification models, first it is necessary to evaluate how accurately these classifiers can categorise landmark images. A collection of training and testing images was collected for this purpose. The training collection was gathered from two sources. The first source was the SUN image dataset [Xiao et al., 2010], which is a large scale collection of images categorised into 899 scene categories. Of these 899 scenes, 7 were deemed useful for the purposes of this work. These included: bridge, building facade, church (outdoor), church (indoor), fountain and statue.

The other source used to gather data for the training set was the Flickr API. For 8 of the 9 semantic categories, the Flickr API was queried using the category name as the query text. All retrieved images were manually analysed and if they conformed to the category class, they were added to the training data. In total the training collection consisted of 3886 images:

Figure 6.1: An example illustrating many different examples in a landmark category 'place of worship'. Although there is a lot of visual intra class variation, it will still be possible, based on visual information alone for many human observers to quickly classify all of these images as either being churches, chapels, cathedrals or mosques.

- Artwork - 246 images
- Bridge - 562 images
- Building - 625 images
- Church - 480 images
- Church (Indoor) - 616 images
- Fountain - 709 images
- Monument - 185 images

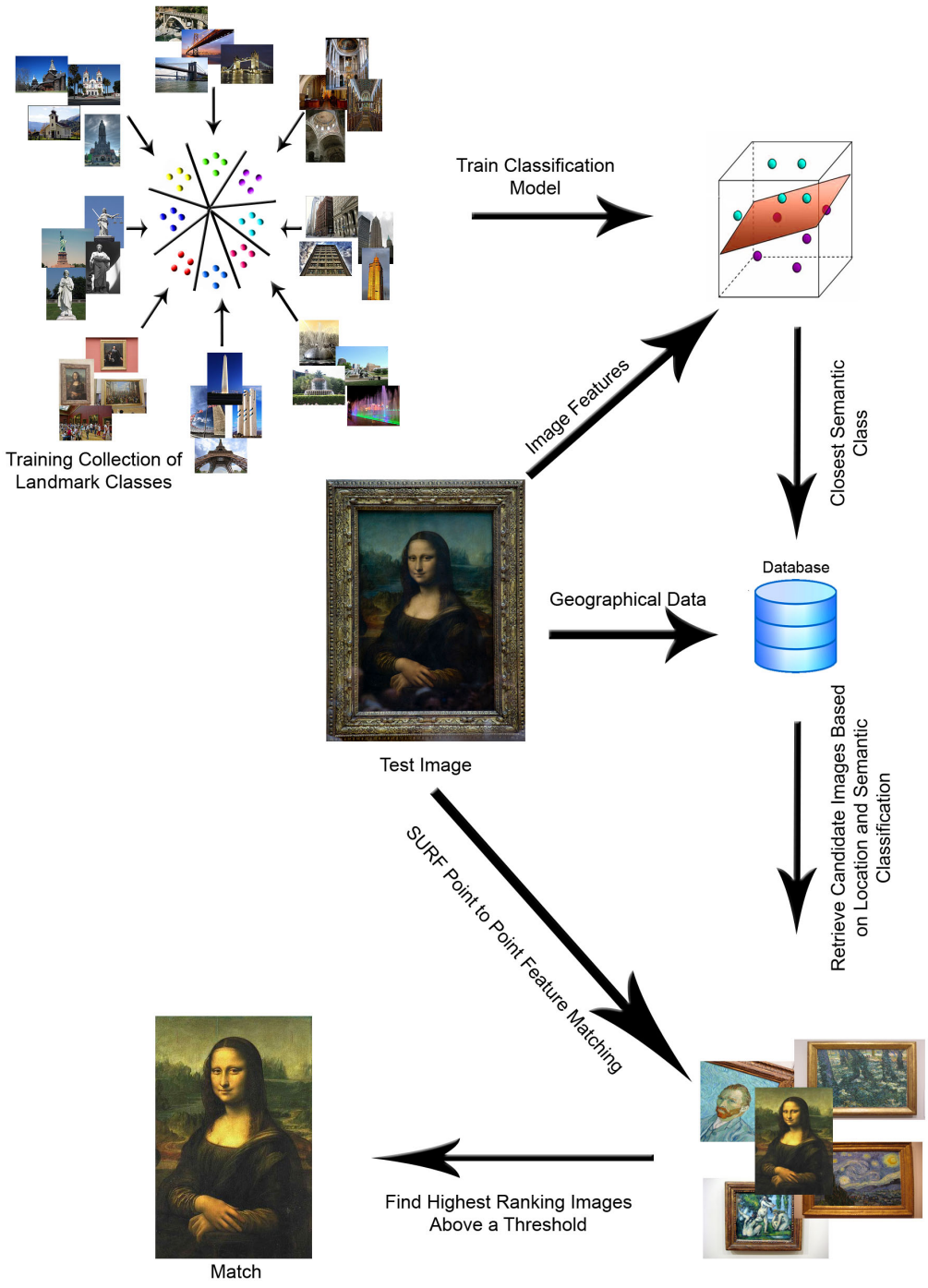

Figure 6.2: An illustration outlining the proposed classification system utilising semantic classification models. For each test image, only images belonging from the same semantic class and a similar location are retrieved from the corpus. SURF interest point matching is then carried out on this smaller subset.

- Other - 155 images

- Statue - 308 images

A multi-class SVM model was trained to classify images into one of these nine categories. Several different kernel functions were evaluated but it was the RBF kernel that performed best for this task. Parameter selection was carried out using *k-fold cross validation* which is described in Chapter 5. An evaluation of this model is described in the next section.

6.3.2 Visual Semantic Classification Evaluation

To evaluate the classifier, a test collection of images was collected. All of these images were retrieved from Flickr using their corresponding landmark class as the query text, with the exception of the 'Other' category. In total for each landmark class 100 images was collected. Each of these images contained geographical data and had been photographed in the Paris region. Insufficient test data could be gathered to evaluate the 'Other' category, therefore, it was not included in the evaluation.

All of the test images were processed through the multi-class semantic classifier with a variety of different input features:

- MPEG7 Edge Histogram
- Visual Bag of Words (Hard Assignment) $k = 1024$, $k = 2048$, $k = 4096$
- Visual Bag of Words (Soft Assignment) $k = 1024$, $k = 2048$, $k = 4096$

The results of this evaluation can be seen in Figure 6.3. If selecting a baseline classification score based on random selection, it would be expected that a correct selection could be achieved around 11% of the time. Therefore, on average the visual classifiers performed significantly better than the baseline.

As expected, some landmark classes could be classified more successfully than others. The highest performing class was 'Church (Indoor)', which achieved an

accuracy score of 88% correct. From informal inspection, the intra class visual variation in this class was deemed to be the lowest across all the classes. The class with the highest level of intra class visual variation, 'Monument' performed very poorly.

A vocabulary size of 2048 performed best for this task. Interestingly, there was a large improvement when using soft assignment as opposed to hard assignment. From these results, it is evident that visual information alone does not allow for an acceptable classification accuracy across all classes.

6.3.3 Landmark Class Classification with Community Created Geographical Data

Visual information can be useful when classifying low-level semantic information from digital images [Szummer and Picard, 1998], however it is more difficult to infer high level semantics. From Figure 6.3, it is evident that global based image features alone are insufficient for accurate classification across all semantic classes. To overcome this, it is hypothesised that utilising geographical contextual information will help to bridge this 'Semantic Gap'. There now exist rich geographical databases, accessible online, that contain high level semantic information describing a specific region. In this section, it is proposed that by fusing visual and geographical information, it would be possible to classify an image into a high level semantic landmark category with a higher degree of accuracy than if using visual information alone.

In recent years, there has been a surge in the creation and dissemination of information on-line by large communities of contributors. One particular type of information accessible online includes geographical data. Large numbers of websites have recently been created that enable for the creation of large scale

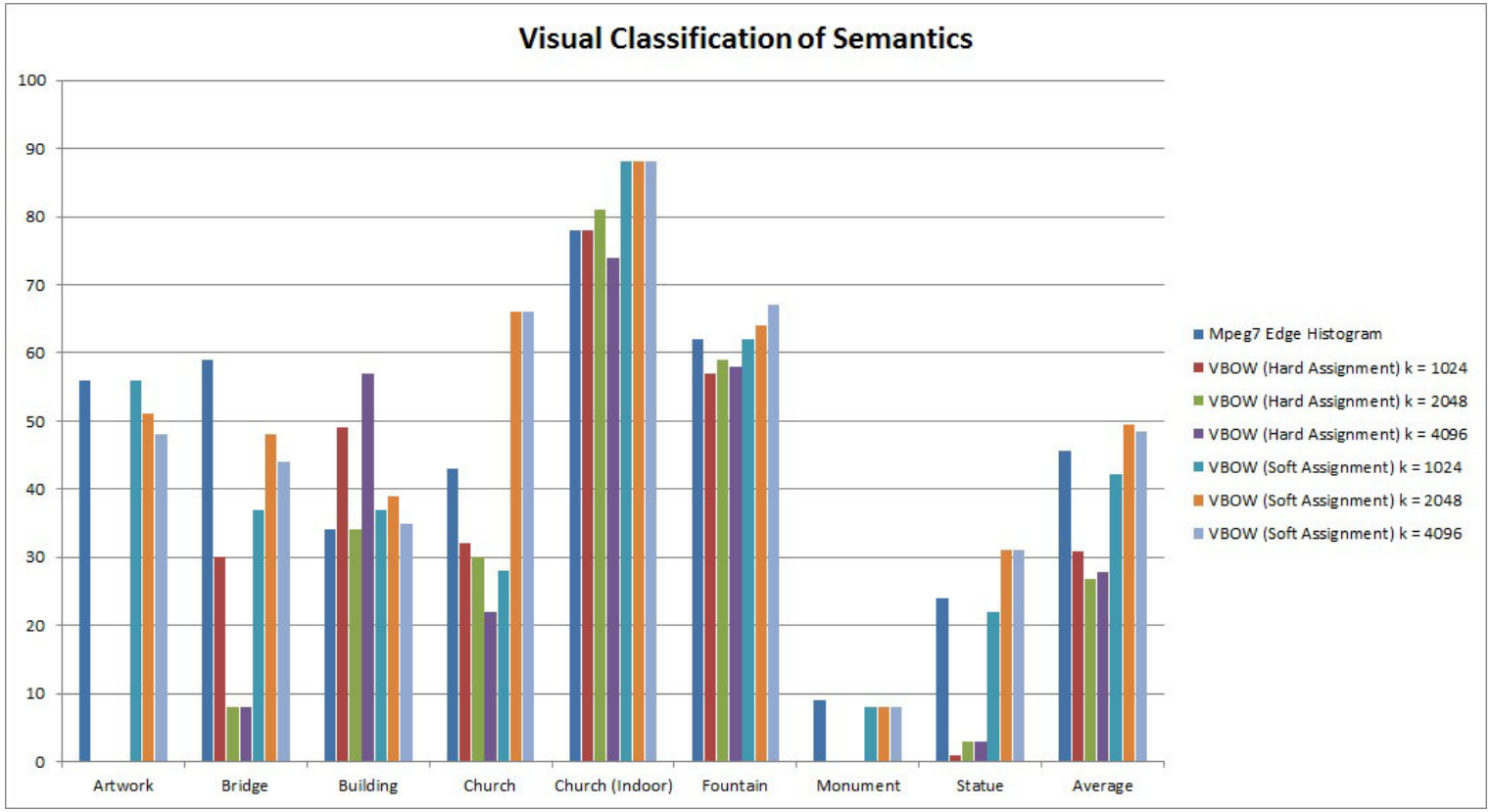

Figure 6.3: The results of the evaluation of the visual semantic classification experiments. An array of input features were evaluated, which as visible to the right of the chart

semantic databases describing geographical locations. One example of these services is 'FourSquare' [FourSqaure, 2009], which allows users to post their geographical location to other users in a social network, and also to provide semantic information about the user's geographical surroundings.

In this work, a database containing geographical points of interest (POI) was created. This consisted of a number of objects referenced by geographical location, which were harvested from two online sources. A technique was proposed to classify an image into one of the 9 landmark categories based on the objects stored in this POI dataset.

6.3.4 Open Street Map

One example of an online geographical community is Open Street Map. Open Street Map is an online repository where community contributors upload the spatial coordinates of a wide range of geographical entities, along with semantic data describing these entities. With a large community of users, these present a very valuable resource for research communities across several fields.

Human contributors can upload map data which is represented by lists of waypoints. Each waypoint contains latitude and longitude coordinates. Users can also upload geographical objects, otherwise known as 'Points of Interest' (POI), and assign them a location. OpenStreetMap has a strict set of guidelines to ensure that uploaded data is accurate. Each uploaded POI can be assigned one of a finite number of feature classes dependent on the use and attributes of the feature. In this work 8 different feature classes were selected to coincide with the set of semantic classes desired to be classified in this work. These feature classes were:

- Bridge
- Building
- Fountain

- Gallery
- Monument
- Museum
- Place of Worship
- Statue

All of these feature classes located in the Paris region were downloaded and stored in the POI dataset.

6.3.5 GeoNames

Another online resource that contains accessible geographical data is the GeoNames repository located at geonames.org. GeoNames is an online geographical repository that contains over 10 million geographically mapped location names, along with 7.5 million geographical features. These features are split into 9 feature classes, which are then split into 645 feature types. Of these feature types, 5 were deemed relevant to the set of landmark classes outlined in section 6.3. Each feature type is associated with a set of metadata, including geographical coordinates, a code representing the country, and the name of the geographical feature. Each of these feature types is considered to be a POI. GeoNames data has been gathered from many reputable sources, including the United States Geological Survey, Netherlands Statistics Office, and the French National Institute of Statistics and Economic Studies, it is therefore expected that this data is quite accurate.

Using the publicly available API, all geographical features located within Paris associated with a set of feature classes was retrieved and stored in a database. This set of feature classes included:

- Bridge

- Building
- Church
- Monument
- Museum

It must be noted that the GeoNames data collection is by no means a comprehensive list for each geographical feature. For several of the features retrieved, the data was quite sparse. For example, for the class Church, only 15 geographical features were found. It must be noted that the majority of geographical features that populate the dataset tend to be well known landmarks, which could be beneficial for this work as these are the objects that users are most likely to visit and photograph. All of these feature classes located in the Paris region were downloaded and stored in the POI dataset. In total, the OpenStreetMap and GeoNames data combined comprised of 1235 POIs.

6.3.6 Evaluation of Classification using Geographical Data

To analyse the effectiveness of community geographical data to classify landmark classes, the test collection of images was processed based on a nearest neighbour scheme. The location information from each test image was extracted and all POIs within a radius of 250 metres were retrieved from the POI database. Retrieved features were then ranked according to geographical distance, using the Haversine formula described in Chapter 3, with the shortest distance ranked at the top. This top ranked feature was then assigned to the test image.

It is assumed that the POIs 'Gallery' and 'Museum' might be useful to classify the semantic class 'Artwork', due to the likelihood of pieces of art appearing in both of these locations. If the closest POI to an 'Artwork' test image is 'Museum' or 'Gallery' then this images is marked as being correctly classified. Similarly for

the semantic class 'Statue', it is assumed that there is a correlation with the POI class 'Museum'. Results from this evaluation are presented in Figure 6.4.

6.3.7 Fusion of Visual and Geographical Features for Semantic Classification

In this section, experiments were carried out that fused the visual and geographical data to ascertain whether a classification accuracy improvement can be achieved using both sets of features. Two fusion approaches were implemented, one based on the presence of a POI in the vicinity of a test image and the other based on the distance between a test image and nearby associated POIs.

Presence of POI Approach

The first fusion technique was based on combining the output values from the SVM classifier with a static value to represent whether a landmark class was present in the POI database. There was no weighted measure applied to this value, and all landmark classes detected within a spatial radius had this value added to its corresponding output from the classifier.

A minor change was made to the libSVM library to output an array of confidence measures C, with a value representing each landmark class $C_1....C_n$ (where n is the number of landmark classes). If the presence of a landmark class was found in the database within a spatial radius of a test image (defined to be 250 metres), a value v was added to c_i, where i is the associated landmark class. Therefore if a POI was discovered in the database associated with the landmark class i then C_i becomes $C_i + v$.

The values in C are normalised into the range 0-1. A value for v is selected based on the maximum value in C, ie. $v = argmax(C)$. Several variations of this calculation were evaluated, some providing a weighted bias towards the visual

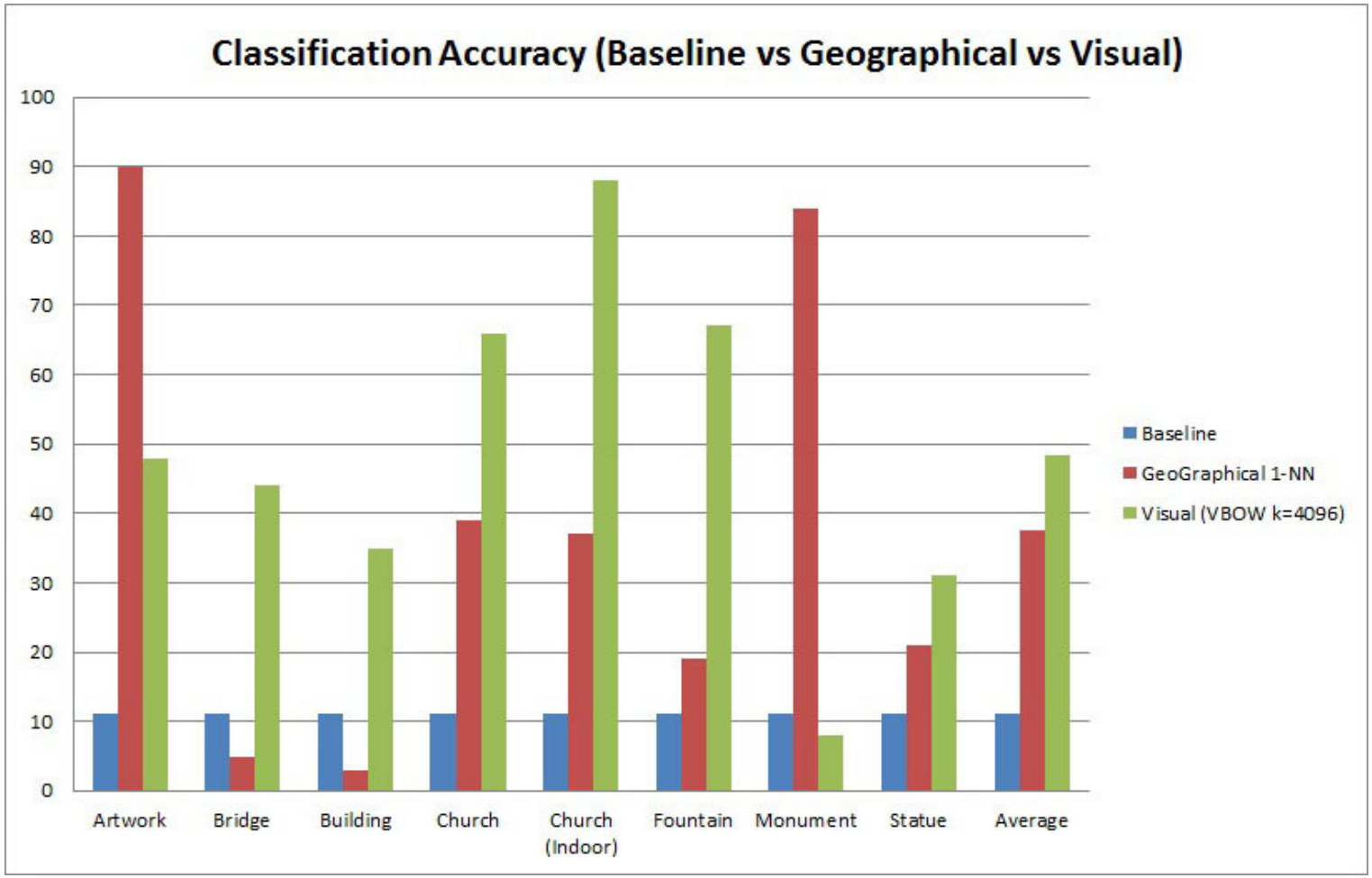

Figure 6.4: A chart comparing the classification accuracy of geographical information and visual information when classifying images into semantic landmark classes. Both are compared against the expected baseline.

data and others providing a weighted bias towards the geographical data. Three weighted variations of $v = w \times argmax(C)$ were evaluated, where w is equal to 2 (denoted as weight 1 in the evaluation), $\frac{1}{2}$ (denoted as weight 2 in the evaluation) and $\frac{1}{4}$ (denoted as weight 3 in the evaluation). A value of 2 for w weights the metric in favour of the geographical data. Values of $\frac{1}{2}$ and $\frac{1}{4}$ for w weight the metric in favour of the visual data.

Weighted Distance Approach

Similarly to the first approach, the second method added a value to relevant confidence measures outputted by the SVM model. The weight of these values was determined by the distance from a POI to the test image. Landmark classes that were nearby had a higher weight assigned to them than those they were located further away. As with above, an array of confidence measures C with a value representing each landmark class $C_1....C_n$ was output from the SVM model.

If the presence of a landmark class i was found in the database within a spatial radius of a test image, a value v was added to C_i. The value v is determined by calculating the distance, denoted as $dist$, between i and a test image t, that was calculated using the Haversine formula described in Chapter 3. Therefore for each POI class that was located within the geographical radius C_i becomes $C_i + (1 - dist(t, i))$ where $dist(t, i)$ is normalised into the range 0 - 1. Four weighted variations of the metric $C_i = C_i + w(1 - dist(t, i))$ were evaluated, where w is equal to 1 (denoted as weight 1 in the evaluation), $\frac{1}{2}$ (denoted as weight 2 in the evaluation), $\frac{1}{4}$ (denoted as weight 3 in the evaluation) and $\frac{1}{8}$ (denoted as weight 4 in the evaluation).

6.3.8 Evaluation

In this section, experiments were carried out to ascertain how accurately a fusion approach (visual and geographical) would perform for the task of landmark class classification. From the results in Figure 6.5, it would seem that the fusion of geographical and visual data for classifying images into semantic landmark categories improves performance slightly over using either visual or geographical features alone for a subset of the landmark classes. On average however, the fusion of visual data with geographical data hinders performance over using visual features alone. The main reason behind this is the sparsity of the geographical database. For many of the landmark classes, there was insufficient data and visual confidence measures were being decreased to the extent that other landmark features that populated the dataset were being incorrectly classified.

To illustrate this point with an example, it can be seen in Figure 6.5, geographical data alone works well for the concept class 'Artwork' but performs very poorly for other concepts, such as 'Building' for example. It would appear that the general poor performance of geographical data is down to the sparsity of the datasets. In the example of the concept 'Artwork', there are very few locations within the city where one would expect to find geo-tagged community images of this concept, possibly less than a dozen (restricted to museums and art galleries). From the geographical data, it can be seen that the largest museum and largest art gallery in Paris(La Louvre and the Musee D'Orsay) are included in the geographical dataset. For the concept 'Building', however, one would expect to find images in a wide variety of locations across the city. Based on this alone, it is logical to assume that the majority of images within the test set of 'Artwork' were geotagged at one of these locations. Additionally, the significant improvement in accuracy that is garnered from the fusion approach over visual features alone for the concept 'Artwork' would imply that with a comprehensive, accurate ge-

ographical dataset, it might be possible to classify all well represented concepts with a high degree of accuracy.

Overall, the accuracy of the scene classification was low and not deemed accurate enough to enable large scale pruning of a search space for use in this framework. Due to the results of the evaluations presented in Tables 6.3, 6.4 and 6.5, the scene classification approach was disregarded and the framework made use of the more accurate vocabulary tree structure approach to classify landmark images that the machine learning based technique failed to recognise.

6.4 Vocabulary Trees

One method to allow for fast approximate nearest neighbour search of image features is the use of a tree structure. One of the first approaches to index visual features into a tree structure for indexing was proposed in [Lowe, 2004]. Lowe suggested using a kd-tree structure to index up to 100,000 SIFT feature point descriptors (approximately 50 images $\times$ 2000 descriptors).

The kd-tree structure is a data structure that stores a finite number of multi-dimensional feature vectors. The structure is a modified binary tree where data is split along the dimension with the highest level of variance. Several variations of the kd-tree have been suggested, their goals however, remain the same, i.e. split a large collection of multi-dimensional points into a finite number of regions so that each region contains the minimum number of points.

The number of nodes in a kd-tree increases exponentially as the number of feature dimensions increases, therefore the kd-tree structure performs very poorly in high dimensional feature space, to the extent that it would perform no better than brute force search ($O(n)$)(in situations where $k > 20$). To overcome this disadvantageous property, Lowe suggest an algorithm that he called Best Bin First (BBF) search. The BBF algorithm is not guaranteed to find the closest neighbour

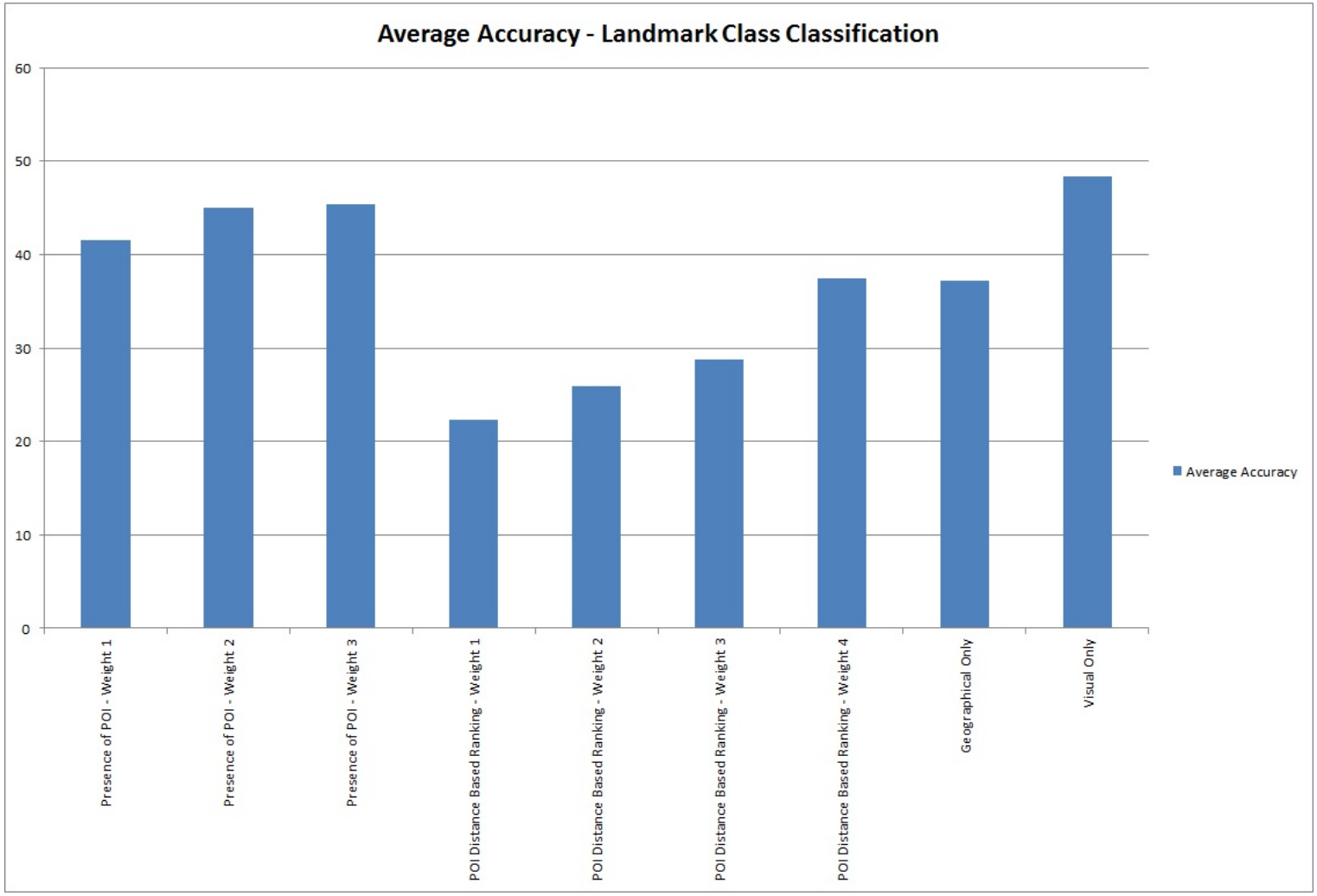

Figure 6.5: A chart comparing the classification accuracy of hybrid approaches to landmark classification against approaches based on geographical and visual information

in the search space, but will approximate a nearest neighbour to within a distance of $1 + \epsilon$. The BBF algorithm is based upon the observation that in any search process, the majority of neighbouring nodes will not contain a nearest neighbour. It selects a number of candidate nodes (eg. 200) that are within the distance $1 + \epsilon$ (where ϵ is a pre-determined parameter) from the query, and limits the search to these candidates. Lowe estimates that 95% of the actual nearest neighbours will be found using this approach. An alternative approach, which provides a simpler approximation model, is based on a hierarchical k-means algorithm and is adopted in this work.

6.4.1 Hierarchical Vocabulary Trees

One alternative approach to kd-trees is the use of a hierarchical k-means tree that was first suggested by Nister [Nister and Stewenius, 2006]. A hierarchical vocabulary tree is a tree structure that similarly to the kd based vocabulary trees is built upon a large visual word vocabulary. It is a form of a hierarchical k-means algorithm, where the inputs consist of visual words, and the clusters centres outputted from each k-means invocation, are used as the pivots of the tree structure.

The algorithm quantises the vocabulary into k smaller subsets at each level using the k-means clustering algorithm on each partition independently. Each quantisation takes place recursively on smaller subsets of data. Instead of the k parameter determining the final number of leaf nodes, k determines the branch factor of the structure. A balanced tree structure consists of a total number of $k \times l$ leaf nodes, where l is the number of levels. An example vocabulary tree structure is presented in Figure 6.6.

Hierarchical vocabulary trees are used in this work, due to their image matching accuracy [Nister and Stewenius, 2006]. Additionally, one advantage of util-

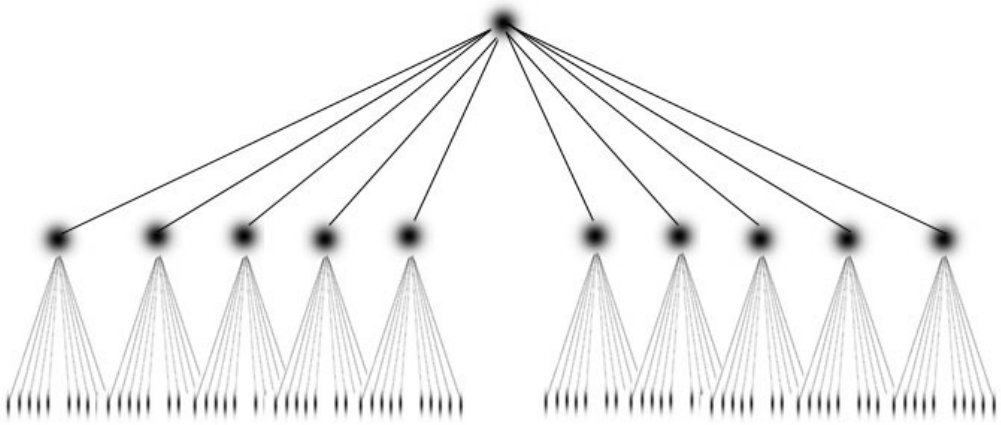

Figure 6.6: An example illustrating the structure of a hierarchical vocabulary tree with a branch factor of 10 and a height of 3 levels.

ising Nister's hierarchical tree approach is that the memory required in a classification system remains static. Once the original tree structure has been built, irrespective of the scale of the corpus, the tree size does not increase. By utilising an effective filtering mechanism, such as geographical based filtering, the discrimination value of the tree structure can be preserved. The hierarchical tree reduces the processing time from O(n) to O(n log k).

To classify a test image, firstly, its spatial data is analysed and only images that are located within a geographical radius of 250 metres are retrieved from the image corpus. Each child node within the tree represents a vocabulary feature, and is given an identification number. SURF features are extracted from the test image and propagated down the tree structure, each feature being assigned an id based on its path down the tree. This list of ids is then compared against the list of ids associated with all retrieved corpus images and identical ids correspond as a match. Corpus images are then ranked based on the number of correspondences, and if that number is above a threshold the corpus image is considered a match.

6.4.2 Hierarchical Tree Evaluation

To ascertain the optimal parameters in using the hierarchical tree approach, several different variations were analysed. In [Lowe, 2004], it is determined that three interest point matches are sufficient to determine a corresponding object, however, as the vocabulary tree approach is based on an approximation technique, it is necessary to determine what number of approximated correspondences constitutes as a legitimate match. It is assumed that this number will be directly related to the number of nodes within the tree. Specifically, when using large vocabulary sizes, the number of tree matches will be quite low, however, as the level of quantization increases, the probability of two non corresponding SURF features following the same path down the tree will increase. In this work, a vocabulary size of 250,000 is chosen to build the vocabulary tree structure.

Several different threshold values were used to determine how many tree correspondences constitutes as a landmark match (5, 15, 25, 35). Additionally two values for the branching factor of the tree, 5 and 10, were also evaluated.

A second evaluation was carried out by utilising SURF point to point matching and re-ranking the already ranked images retrieved using the vocabulary tree approach. This was carried out as a confirmation stage to eliminate false positives. As point to point matching is an expensive process, two approaches were implemented. The first approach carried out SURF re-ranking on all retrieved images from the vocabulary tree, while the second approach carried out SURF re-ranking only on the top k ranked images to reduce processing time. It is expected that these images are more likely to be correct matches, thus this approach eliminates potentially large numbers of unnecessary image comparisons. Geometrical verification is not used at this stage to reduce the processing time required and to maintain real time recognition. Additionally, it has been shown that geometric verification in similar landmark recognition tasks using a hierarchical vocabulary

tree produces a relatively minor precision increase when using large vocabulary sizes (6% with k = 250,000) [Philbin et al., 2007] (it must be noted that when using smaller vocabularies (50k), there was a significant improvement in precision scores). The results of these evaluations can be see in Tables 6.1 - 6.5.

6.5 Hybrid Approach to Landmark Recognition

In this section, a hybrid approach to landmark classification is introduced and evaluated. This hybrid framework combines the machine learning based approach described in Chapter 5 with the vocabulary tree based approached described in section 6.2. This hybrid framework takes advantage of the fact that a large proportion of a training collection of landmark images for a city will contain many large clusters of near identical imagery for a number of commonly photographed landmarks. Instead of matching against all images in a training set located within a certain region, the hybrid approach will quickly ascertain whether a test image contains one of the more commonly photographed landmarks within the region.

A slight decrease in processing time is achieved by using machine learning approaches, as described in Chapter 5, over the vocabulary tree approach described in section 6.2. The hybrid method takes advantage of this by first attempting to classify an image using the machine learning models. If the number of matches is beneath a threshold, the test image is then processed using the vocabulary tree based technique. Due to the noisy textual metadata in the corpus, it is deemed that a minimum of three images are required to be retrieved for a query image, for that query image to be considered matched. This is to compensate for the inaccuracies in the textual metadata. It is more robust to find relevant tags, with which to annotate a query image, from a number of images as opposed to just one. Work concentrating on tag selection schemes based on query matches is described in Chapter 7.

An outline of the hybrid system is illustrated in Figure 6.7. The hybrid approach consisted of the top performing machine learning method which was using a VBOW feature with a vocabulary size of 4096. This method used an SVM model based on the Radial Basis Function kernel.

Several variations of the hybrid method were evaluated utilising different vocabulary tree attributes. The evaluated hybrid variations were as follows:

1. SVM + vocabulary tree (correspondence threshold of 25) with no SURF feature point re-ranking
2. SVM + vocabulary tree (correspondence threshold of 25) with SURF feature point re-ranking
3. SVM + vocabulary tree with SURF re-ranking restricted to top 20 nearest neighbours
4. SVM + vocabulary tree with SURF re-ranking restricted to top 30 nearest neighbours
5. SVM + vocabulary tree with SURF re-ranking restricted to top 50 nearest neighbours

The results of this evaluation can be seen in Table 6.6.

6.6 State of the Art Techniques

6.6.1 Landmark Categorisation using Inverted BOW Indexes

In recent years, the computer vision community has been very active in developing methods to aid image matching operations in very large scale image collections. In the past few years, problems in this domain that were previously thought intractable have been made possible with advances in computer vision

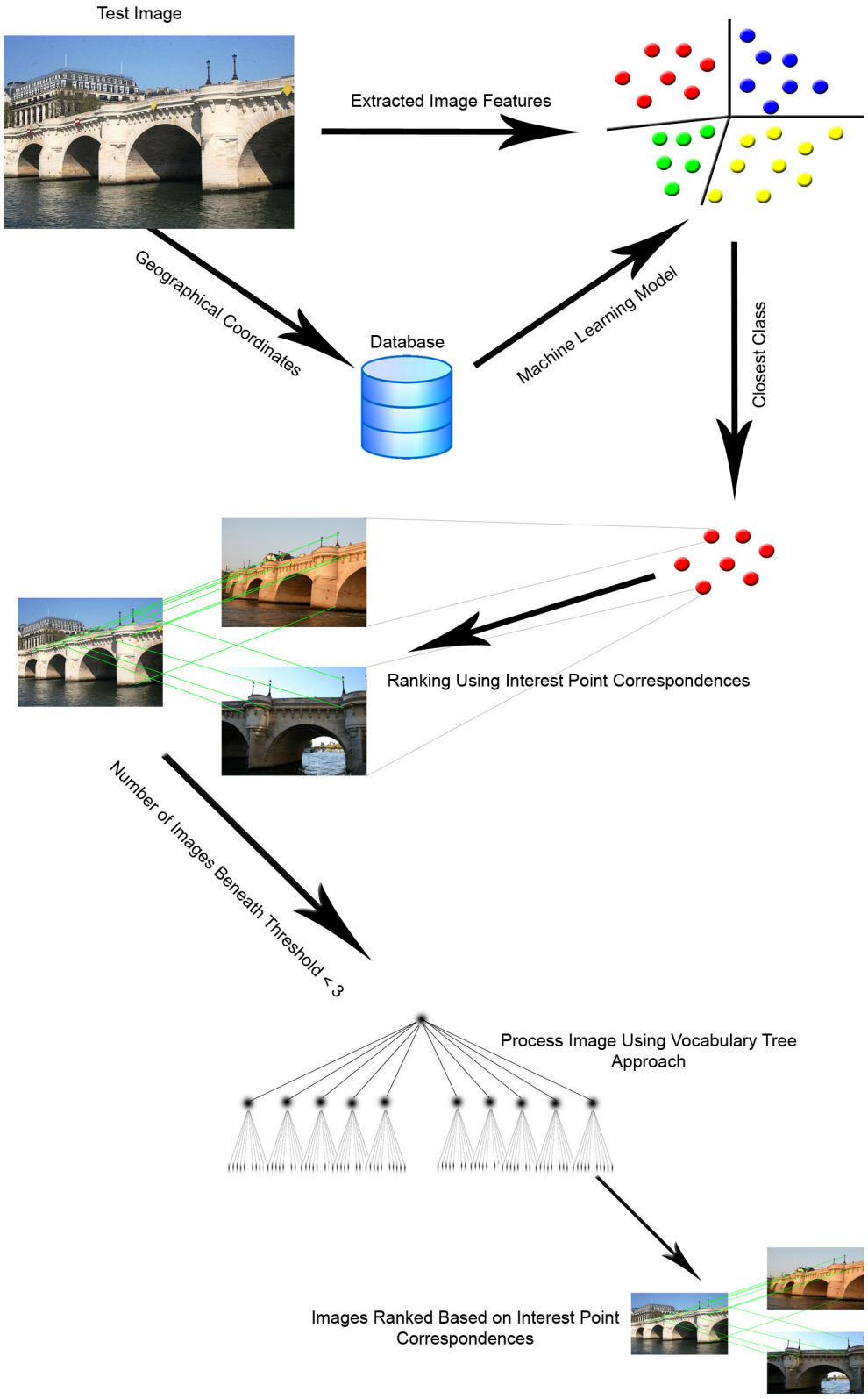

Figure 6.7: An example illustrating the structure of a hierarchical vocabulary tree with a branch factor of 10 and a height of 3 levels.

and image retrieval algorithms, along with many advances in several related fields. There now exist many established techniques to achieve relatively accurate image matching and retrieval, even from within datasets consisting of millions of images based solely on automated approaches.

One of the most commonly used methods to achieve this is based on the visual word paradigm merged with a commonly used text retrieval technique for efficiently storing large amounts of textual information. The files in which this data is stored are called 'inverted files'. First introduced by Sivic et al. [Sivic and Zisserman, 2003] for the purposes of finding similar video frames in motion pictures, the inverted visual word technique has been shown to perform very well in a wide variety of image matching applications in a myriad of domains, including landmark recognition [Philbin et al., 2007]. This technique has also been implemented into commercial landmark image recognition systems such as Google Goggles [Goggles, 2008], which is a mobile phone based image recognition system that allows for a user to photograph an object or scene and the application will attempt to classify the image, along with providing information about the main object depicted in that image.

The inverted visual word approach is very similar to the vocabulary tree method described in this chapter, however, there are a few key differences.

Inverted Index Files

The main idea behind inverted visual word features is to adopt a text retrieval style approach to image retrieval. Inverted indexes have been used to efficiently store and describe a collection of documents in information retrieval [Baeza-Yates and Ribeiro-Neto, 1999]. Inverted indexes based on visual vocabularies follow the same approach. As opposed to storing a set of detected visual words for each individual corpus image, a list is stored for each visual word within the vocabulary. This list contains identification numbers representing each

image within the corpus in which that word was detected. Query images are then matched against these inverted lists. This indexing method reduces the space required to store features in memory and speeds up the matching process.

Term Weighting

When comparing image features using a large vocabulary, some features will be more discriminate than others. For example, a vocabulary feature that is detected in 1% of all images within a corpus will not have the same discrimination value as a feature that occurs in 0.01% of the corpus. To account for this, a method was proposed in [Sivic and Zisserman, 2003], that takes into account the discrimination value of each visual word feature based on frequencies. A tf-idf [Sparck Jones, 1988] approach is used to assign a weight to each visual word in the vocabulary.

Geometric Ranking

Once a ranked list of relevant images have been retrieved, they are re-ranked based on geometric consistency. In [Philbin et al., 2007], a variant of the RANSAC algorithm, called LO-RANSAC [Chum et al., 2003], is used to measure geometric consistency. This is an approximation of the RANSAC algorithm used to improve classification time. In this work, the standard RANSAC algorithm is used as described in Chapter 4, to re-rank the images, which should aid precision.

Evaluation

In this section an inverted visual word approach based on the work in [Philbin et al., 2007] was implemented. The goal is to evaluate the approaches in this work against this widely used state of the art method. In the evaluation section, this approach is referred to as inverted indexes of visual words.

6.7 Evaluation of Hierarchical Matching

To evaluate the vocabulary tree and hybrid approaches, the test collection used in the evaluation in Chapter 5 was used. The same evaluation metrics as Chapter 5, were also used. These consisted of:

- Precision (Average)
- Image Recall
- Image Recall (Relevant)
- Precision Top 3 - Precision(3)
- Precision Top 5 - Precision(5)
- Precision Top 10 - Precision(10)
- Recall
- Classification Time

6.7.1 Vocabulary Tree

The vocabulary tree approach to classifying landmark images is evaluated in this section. Experiments were carried out with 2 values for k, 5 and 10.

The branch factor parameter led to a large difference in the overall performance of the tree. The tree with a branch factor of 5, achieved an increase in precision of approximately .06 when using a threshold score of 15. This is due to the increase in the number of tree levels and therefore an image feature is compared against a smaller number of k-means values a larger number of times. A branch factor value of 10 led to a slight decrease in precision but a large increase in image recall. In the SURF feature re-ranking stage there was a larger number of relevant images in the candidate set and therefore a higher precision was achieved.

Additionally, 4 different threshold values were analysed to determine a matched image. These values were 5, 15, 25 and 35. It can be assumed from analysing classification times in table 6.2, that when a low threshold is assigned to the acceptable number of tree correspondences, as expected, the number of candidate images retrieved grows significantly. As can be seen from Table 6.1, with a threshold value of 5, the tree based approach actually achieves an image recall (relevant) score of 1.0 (i.e. every image in the test collection had at least 1 relevant match). However, with this threshold value the precision performance is very poor. With the highest evaluated threshold value of 35, the precision score increases by more than 200%. Additionally, when using a high threshold the image recall performance is quite poor. Of all the evaluated threshold values, the most encouraging is a value of 15.

A SURF based image re-ranking scheme significantly improves the precision score of the vocabulary tree approach. As can be seen from Tables 6.2 and 6.4, adding a SURF re-ranking stage can improve precision by up to 50%. This is particularly important when a low-threshold score was utilised to represent an image match using tree correspondences. With a high threshold the improvement is less obvious but still significant. The inverted index approach outperforms the standard vocabulary tree method, however, when SURF feature ranking is carried out on the output from the vocabulary tree, the precision significantly outperforms the inverted index method, as illustrated in Figure 6.8.

The main issue with using a low threshold value of 5 is that a large number of images pass the threshold and the SURF re-ranking stage requires an unacceptable amount of processing time. When using a threshold value of 5, it requires over 3 minutes to process a test image. Clearly, this is an unacceptable time frame. To address this issue, only the top k ranking images output from the vocabulary tree are processed by the SURF re-ranking scheme. 3 values for k were analysed 20, 30 and 50. It can be seen from Table 3.5 that this approach yielded a significant improvement in terms of classification time and precision.

Match Threshold	5	15	25	35
Precision(Overall)	0.319	0.514	0.612	0.697
Image Recall	.963	.688	.453	.330
Image Recall (Relevant)	1.0	.830	.547	.398
Precision(3)	0.695	0.778	0.822	0.851
Precision(5)	0.595	0.711	0.774	0.814
Precision(10)	0.493	0.647	.0.720	0.776
Recall	.17	.13	.12	.11
Classification Time (ms)	2225	2210	2188	2183

Table 6.1: Classification results: Hierarchical Vocabulary Tree (Branch Factor = 5)

Match Threshold	5	15	25	35
Precision(Overall)	0.875	0.936	0.925	0.937
Image Recall	.505	.490	.332	.247
Image Recall (Relevant)	.609	.507	.400	.298
Precision(3)	0.961	0.989	0.988	0.996
Precision(5)	0.934	0.978	0.981	0.984
Precision(10)	0.905	0.955	.0.968	0.970
Recall	.31	.20	.22	.21
Classification Time (ms)	173500	40122	5300	3139

Table 6.2: Classification Results: Hierarchical Vocabulary Tree with SURF Correspondence Re-Ranking(Branch Factor = 5)

6.8 Hybrid Evaluation

In this section, a hybrid approach was evaluated that consisted of an SVM based approach using a value of 4096 for k and the RBF kernel (These were the parameters that were deemed optimal from the evaluation in Chapter 5), and a vocabulary tree. 5 different hybrid approaches were evaluated:

Match Threshold	5	15	25	35
Precision(Overall)	0.302	0.457	0.575	0.636
Image Recall	.975	.767	.531	.382
Image Recall (Relevant)	1.0	.926	.641	.461
Precision(3)	0.688	0.744	0.801	0.825
Precision(5)	0.583	0.670	0.738	0.768
Precision(10)	0.487	0.604	0.684	0.727
Recall	.18	.13	.12	.12
Classification Time (ms)	2160	2169	2164	2160

Table 6.3: Classification results: Hierarchical Vocabulary Tree (Branch Factor = 10)

Match Threshold	5	15	25	35
Precision(Overall)	0.874	0.876	0.918	0.947
Image Recall	.780	.529	.371	.281
Image Recall (Relevant)	.942	.638	.448	.339
Precision(3)	0.965	0.980	0.994	0.981
Precision(5)	0.940	0.970	0.984	0.974
Precision(10)	0.908	0.950	.0.966	0.964
Recall	.34	.305	.24	.17
Classification Time (ms)	183943	47102	7300	3403

Table 6.4: Classification Results: Hierarchical Vocabulary Tree with SURF Correspondence Re-Ranking(Branch Factor = 10)

Top k images	$k = 20$	$k = 30$	$k = 50$
Precision(Overall)	0.900	0.866	.85
Image Recall	.660	678	.703
Image Recall (Relevant)	.797	818	.849
Precision(3)	0.966	0.926	.916
Precision(5)	0.949	0.910	.900
Precision(10)	0.923	0.900	.887
Recall	.18	.17	.18
Classification Time (ms)	6105	8118	11553

Table 6.5: Classification results: Hierarchical Vocabulary Tree - SURF correspondence matching - Top k Images

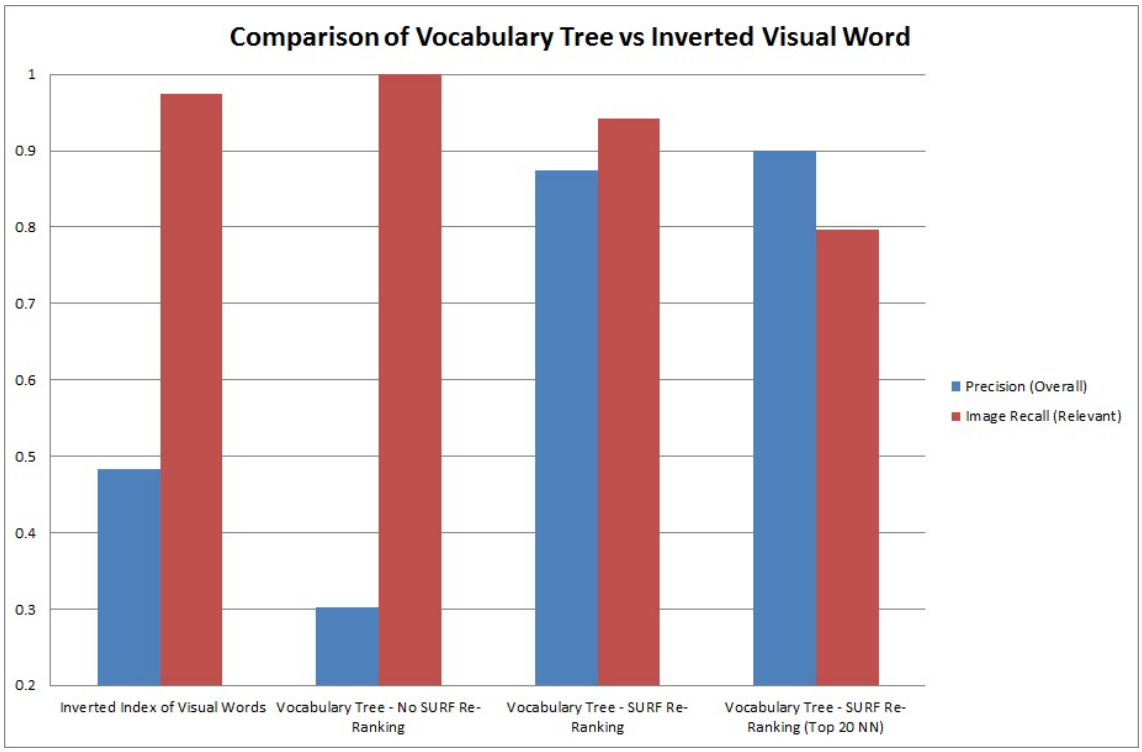

Figure 6.8: A graph outlining the precision and image recall (relevant) scores achieved by several of the evaluated vocabulary tree approaches

1. SVM + vocabulary tree (correspondence threshold of 25) with no SURF feature point re-ranking

2. SVM + vocabulary tree (correspondence threshold of 25) with SURF feature point re-ranking

3. SVM + vocabulary tree with SURF re-ranking restricted to top 20 nearest neighbours

4. SVM + vocabulary tree with SURF re-ranking restricted to top 30 nearest neighbours

5. SVM + vocabulary tree with SURF re-ranking restricted to top 50 nearest neighbours

The results of the hybrid approach evaluation are presented in Table 6.6. It is evident from these results that the hybrid approach has some desirable attributes

over using either of the SVM or vocabulary tree approaches alone. It achieves an increase in precision and image recall (relevant) scores over the SVM approach by a large margin. Additionally it achieves a significant decrease in processing time over using the vocabulary tree approach. The image recall score is also improved by using the hybrid approach over using a vocabulary tree alone.

As expected, when using the hybrid method without SURF re-ranking, the precision was the lowest out of all evaluated approaches. The vocabulary tree alone does not provide a sufficient level of discrimination and a re-ranking stage is required to rectify this. It might however, be possible to improve precision by enlarging the vocabulary size. The smaller the level of quantisation, the more discriminative the power of the vocabulary tree approach. By enlarging the size of the vocabulary, the memory requirements would increase but still remain static.

Of all evaluated hybrid methods, it was number 3 that performed optimally. It achieved the highest level of precision which is deemed to be the most important attribute. It is deemed that a minimum of 3 retrieved images is required to provide an annotation for a test image (described in Chapter 7) due to inaccuracies in the metadata. When measuring the precision value for the top 3 retrieved images using the optimal hybrid approach, it was found that a score of .916 could be achieved. This is a very encouraging result, which means that over 91% of all images were classified correctly.

It can also be seen from the results, the hybrid approach improves upon the vocabulary tree approach in terms of image recall and required processing time. This is depicted in Figures 6.9 and 6.10. The hybrid method achieves a higher image recall (relevant) score by a value of .056. This improved image recall is because a number of test images are classified by the SVM based method that were missed using the vocabulary tree. There is a slight decrease in precision using the hybrid approach, which is due to the lower precision scores being output from the SVM models.

Hybrid Approach	1	2	3	4	5
Precision(Overall)	0.601	0.622	**0.809**	0.794	.786
Image Recall	.688	.559	**.707**	.721	740
Image Recall (Relevant)	.830	.675	**.853**	.870	.893
Precision(3)	0.818	0.906	**0.916**	0.902	0.895
Precision(5)	0.758	0.875	**0.889**	0.876	0.869
Precision(10)	0.688	0.828	**0.847**	0.841	0.833
Recall	.08	.10	**.17**	.16	.13
Classification Time (ms)	3176	4774	**5139**	5788	7541

Table 6.6: Classification results: Hybrid Approach

When the hybrid method is compared against the state of the art approach, it shows encouraging results. It achieves a significantly higher precision score, while sacrificing a slight decrease in image recall. This is depicted in Figure 6.11. The state of the art method requires a significantly lower processing time than the hybrid, however, it still achieves recognition in real-time. Overall, the results of this evaluation show that the hybrid approach allows for accurate recognition in real-time and compares favourably against state of the art techniques.

6.9 Conclusions

In this chapter, two approaches to classifying uncommon viewpoints of landmarks were evaluated. The first approach adopted a scene classification method with the aim of reducing the search space as part of a nearest neighbour search problem. It was demonstrated that this approached performed poorly. It would seem that visual features alone lack the required discrimination power to differentiate between visually similar classes of objects. To address this issue, a geographical database containing instances of each of these landmark classes was collected from online sources. This data showed encouraging results in situations where the

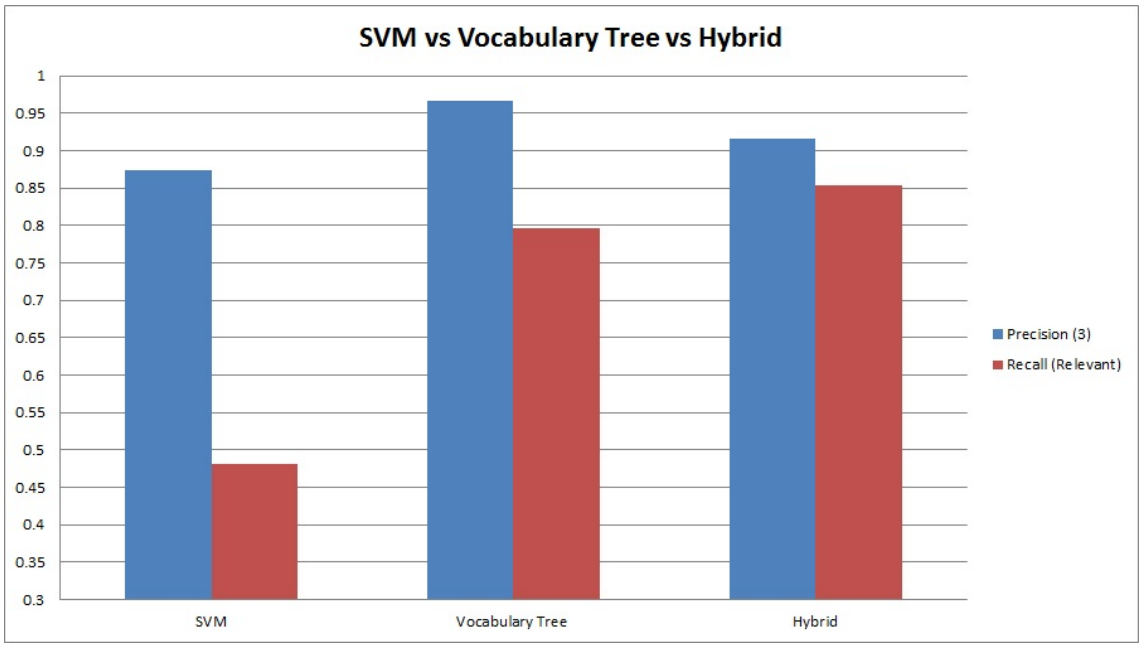

Figure 6.9: A diagram illustrating the precision and image recall (relevant) scores for each of the evaluated systems: SVM, Vocabulary Tree and Hybrid

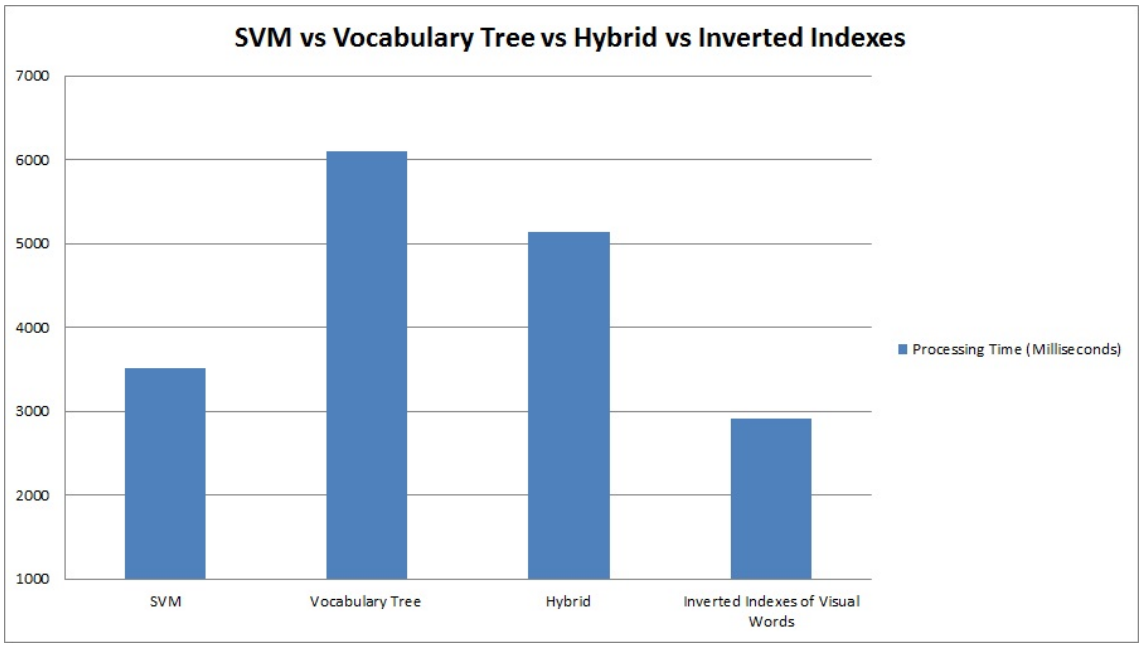

Figure 6.10: An chart illustrating the processing time required for each of the evaluated approaches. The time is measure in milliseconds.

landmark class was represented correctly in the data, for other classes however, the data set was too sparse to gain any advantage from its use, and in fact, decreased classification accuracy. Based on the evaluation, it is assumed that

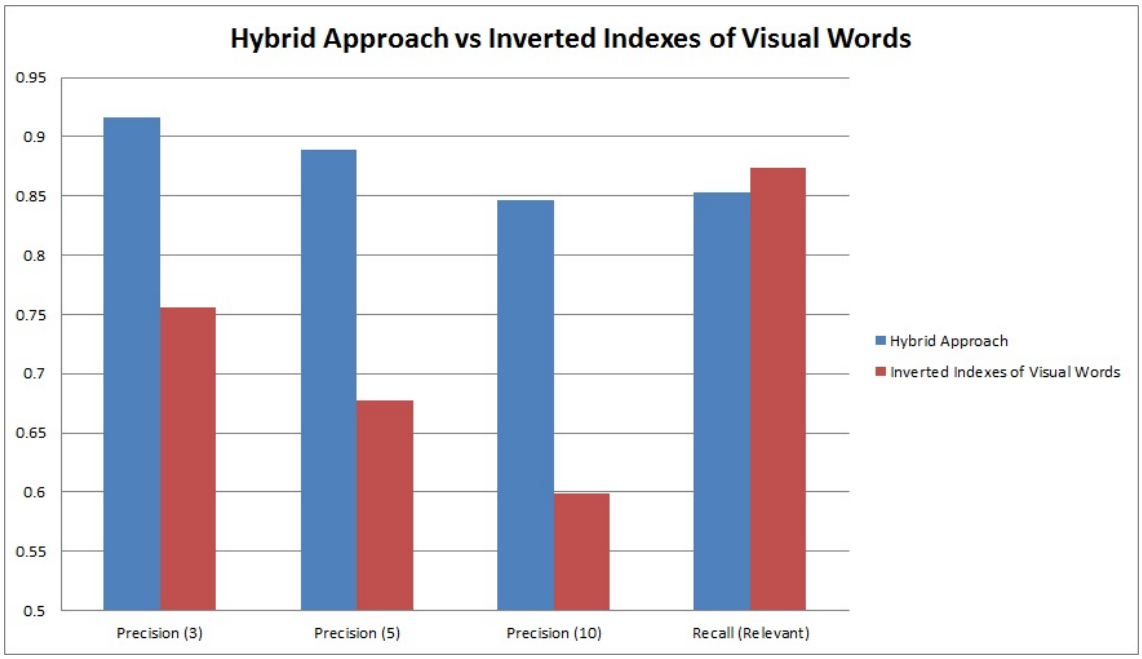

Figure 6.11: A chart comparing the hybrid approach to the inverted index method suggested in [Philbin et al., 2007]. The compared metrics are precision(3),precision(5), precision(10) and image recall (relevant)

with a large scale geographical dataset containing a larger representation of many of the landmark classes that it might be possible to accurately group landmark images into categories semantically.

The second approach evaluated was a method based on vocabulary tree structures to approximate nearest neighbour matches of interest points in O(log n) time. It was determined that a significant precision increase can be achieved by using a two stage approach consisting of adding a SURF re-ranking scheme to the output ranked lists of test images from the vocabulary tree. It was also found that by restricting this SURF re-ranking process to the top k images, there was an increase in precision and processing time requirements.

In this chapter, it was also demonstrated that by fusing the SVM based approach described in Chapter 5 with the vocabulary tree method, there was an improvement on using solely the vocabulary tree method, in terms of processing time and the image recall (relevant) metric. This method sacrificed a small

reduction in precision but this was minimal. This hybrid approach was compared against a commonly used landmark recognition approach [Philbin et al., 2007] and achieved encouraging results. The hybrid method outperformed the inverted index approach by a large margin in terms of precision. However, it must be noted that these precision increases are measured against the benchmark and might be slightly skewed as the image features using to calculate the benchmark are similar to those used to rank images in the hybrid approach.

Overall, the hybrid approach has achieved some very encouraging evaluation results. The processing time required over brute force search has been reduced by a factor of over 100, which represents a significant improvement. It allows for the classification of a landmark image with a precision of over .91 when analysing the top 3 retrieved images which means that there is only a decrease of approximately 9% in classification accuracy. Another important improvement with the use of this hybrid approach over other commonly used methodologies, is that there is a static memory requirement (in this work, deemed to be approximately 400MB), which combined with geographical information will allow for scaling this approach up to a much larger training corpus, without any memory restrictions.

Chapter 7

Selecting Relevant Annotations from Community Metadata

7.1 Introduction

Many of the large scale online photo repositories such as Flickr [Flickr, 2004] use text based retrieval methods to return images relevant to a users search query. In order for this approach to work, it is necessary for a user to manually create a textual description describing the content or context of an image. There is a need to automate this procedure as many users will not spend sufficient time required to carry out this task. The aim of this chapter is to create an automated method to associate a set of metadata with a query image based on the output of the image recognition framework.

If an appropriate match is found for a query image, the landmark recognition framework will return a ranked list containing a number of images, which in this chapter will be referred to as the result set., each of which has a number of tags associated with it. The aim of this chapter is to evaluate techniques to extract rich semantic information from these retrieved tags and associate that information with the query image. Due to the noisy nature of the data, there is a considerable

challenge in selecting from this set of potential annotations one or more tags that have a high semantic relevance to a query image. It would be pointless to successfully recognise and classify a landmark within an image only to annotate that landmark with a set of heterogeneous or inaccurate text annotations.

In recent years, work has been carried out analysing how best to extract representative tags from clusters of images in community contributed datasets. Kennedy et al. [Kennedy et al., 2007] explored different methods to structure Flickr data, and to extract meaningful patterns from this data. Specifically, they were interested in selecting metadata from image collections that might best describe a geographical region. In similar work [Kennedy and Naaman, 2008], focused these techniques on extracting textual descriptions of geographical features, specifically landmarks, from large collections of Flickr metadata. Tags are clustered based on location, and using a tf-idf approach tags are selected intended to correlate with nearby landmarks.

Ahern et al. [Ahern et al., 2007] employ a tf-idf approach on sets of Flickr tags to create a visualisation of representative tags overlayed on a geographical map. They call this system the 'World Explorer', and it allows users to view unstructured textual tags in a geographically structured manner.

Xirong et al. [Li et al., 2009a] combine visual information with a tf-idf scoring metric to estimate tag relevance within a dataset of Flickr images. For each test image, they carry out a visual search procedure to find its nearest neighbours visually within the dataset. They show that by calculating co-occurrences of tags within visually similar images, it is possible to estimate relevant tags for a query image over using text based methods alone with a higher probability.

Most of the approaches to date have focused on variations of text-retrieval based models using a tf-idf [Sparck Jones, 1988] scoring approach to choose relevant representative tags from a cluster of metadata [Mahapatra et al., 2011]. In this chapter, the aim is to improve on this work, by analysing alternative statisti-

cal methods, using other sources of information that are accessible through the recognition framework described in this thesis and within the community data itself.

7.2 Tag Selection Schemes

There is a significant challenge in retrieving semantically relevant annotations from this dataset, as it is known from Chapter 3 that much of the data is heterogeneous and semantically non relevant. An example of the set of metadata retrieved for a test image is depicted in Figure 7.1. In this section, many different approaches are proposed to solve this problem. The goal is to create a method that will optimally select relevant tags for an image that replicate those that might be selected by a human annotator. For any query image, particularly if that image is depicting a popular landmark from a commonly photographed viewpoint, there could be a large number of matched images returned from the image corpus. Each of these retrieved images will have its own set of textual tags, with no guarantee that any tag is relevant. It remains a challenge, when captioning query images, to select a relevant tag or set of tags that might best semantically describe the landmark depicted within the query image.

In this work, a number of tag selection schemes were implemented and evaluated. From the structure of the data, three different types of selection schemes were identified.

- **Frequency Based** schemes use information that is available regarding the frequency of individual tags within a result set and across the entire corpus. It is hypothesised that tags with a correlation across a high number of images within a result set have a higher probability of being relevant to a test image as opposed to those with a low correlation. Additionally, it is assumed that

a tag with a high frequency across the entire corpus is less likely to have a high degree of semantic relevance to a test image.

- **Ranking Based** schemes utilise information that is provided from the structure of the data output by the framework that is described in this thesis. Images within a result set are retrieved with a rank describing how visually similar each image is to the query image. It is assumed that a more visually similar image has a higher probability of having relevant metadata than that of a less visually similar image. A second set of experiments are carried out regarding the ranking of each tag and its associated image.

- **Geographical Based** schemes take advantage of the geographical information that is associated with each image and its associated tags. It is assumed that tags with a large distribution across a large region would be less likely to be describing an individual landmark, and would be more generic than tags with a small spatial variance.

7.2.1 Tag Selection Based on Term Frequency

The first approach evaluated was based on selecting the tag with the highest term frequency score within a result set. Term frequency (TF) is calculated by the number of times a tag appears within a result set, divided by the total number of images within the result set. Tags were ranked based on descending term frequency scores, which essentially corresponds to the terms with majority representation within a result set at the top of the ranking.

Although it would seem intuitive that the tags with the highest frequency are considered the most representative tags of the result set, and in many cases they are, several problems exist with this approach. From empirical inspection, it seems that generic tags such as 'Paris' and 'France' are regularly the tags with

paris,**france**,church,window,glass,**cathedral**, **seine**, military, **architecture**, canoneos400d,
notredame, **europe**,eos,rebel,
street, **city**, kathedrale,kirche,**trip**,**travel**,winter,**vacation**,history,
architecture,french,education,view,military,cité,capital,gothic,**sightseeing**,culture,
tourist,christian,**gargoyle**,abroad,bible,christianity,2008,continent,europeanunion,overseas,
latinquarter,buttress,arrondisement,colonialism, **rossette**, **stainedglass**, faith,
rio,río,river,**french**,francia,parís,sena,
verano,francia,vacations,vacaciones,**eglise**,historia,fé,vitral,gotico,vitrales,xti,betarouge,
stainglasswindow,cathédralenotredamedeparis,

Figure 7.1: An example illustrating the wide range of relevant and non-relevant tags that have been retrieved from images matched with the test image above. Each tag is sized according to its frequency within the result set of matched images. A tag with a higher frequency is depicted larger than a tag with a low frequency.

highest frequency scores. While annotating a query image with the generic tag 'Paris' could be useful in certain circumstances, it still has a low value semantic meaning in this dataset, and therefore a low discrimination value. It is already possible to ascertain from the geographical information that the image was located in Paris, so the visual matching process adds no valued additional information.

The tf score was calculated by using the following formula:

$$\text{tf}_i = \frac{t_i}{q}$$

where t_i is the number of times a tag i appears within a result set, and q represents the total number of images within the result set.

7.2.2 Tag Selection Based on Global Frequency Distributions

The main problem with using an approach based on term frequency is that all of the tags retrieved are considered to be of equal importance, and have an equal probability of being ranked highly regardless of the tag's discriminating power. One important measurement in determining the importance of a candidate tag is its level of 'uniqueness' or 'specificity' (as defined in [Sparck Jones, 1988]) across the entire corpus.

Following a document retrieval methodology, a method based on the 'term frequency - inverse document frequency' (tf-idf) frequency approach is implemented. This method assigns a higher score to tags that have a high term frequency within a result set, and a lower frequency across the entire corpus. Additionally, it will assign a low score to any tag that occurs regularly across the corpus.

The tf-idf metric is a combination of the term frequency metric defined in section 7.2.1, and a metric called the inverse document frequency (idf). The document frequency of a tag t is defined as the number of images within the corpus that contain t. To scale the weight of the document frequency, an inverse document frequency of a tag is defined as

$$idf_t = \log \frac{df_i}{N}$$

where df_i is the document frequency of the tag i and N is the total number of images within the corpus. The tf-idf metric is then formulated as:

$$tf\text{-}idf_t = tf_t \times idf_t$$

Each tag within an image result set was assigned a tf-idf score using this metric, and tags were ranked in a descending order with the top k tags selected as the most representative or relevant.

7.2.3 Tag Selection Based on Image Similarity Rankings

Each result set is ranked based on visual similarity to the query image, with the highest ranking images having the highest number of SURF correspondences. It would seem logical to analyse whether this visual relationship with an image corresponds to contextual similarity within the associated tags. The higher the rank of an image, the more likely it is that the image is a correct match. An incorrectly matched image is more likely to contain irrelevant tags, therefore it seems plausible that the higher ranked images have a higher probability of containing relevant tags. To evaluate this hypothesis, a tag selection scheme based on the ranked position of each matched image was carried out. The higher the rank of an image, the larger the weight associated with its corresponding tags. Two weighting schemes were implemented, both of which were based on a mixture of tag frequency within a result set and image ranking.

The first is similar to the weighting scheme adopted in Chapter 5 for the soft assignment of visual words. This scheme places a large importance on a small number of high ranked images, while the weight associated with images lower down the ranking system is decremented significantly, to such an extent that the lowest ranked images are effectively irrelevant. The score assigned to each tag t was calculated as follows:

$$score(t) = tf_i \times w_i \quad \text{where} \quad w_i = \frac{1}{r}$$

where n is the total number of images within the result set that the tag t appears in and r is the rank of the image i.

The second ranking based scheme provided a more balanced weight across all ranked images. The weight associated with lower ranked images is decremented more slowly. This scheme can be formulated as:

$$Score(t) = tf_i \times w_i \quad \text{where} \quad w_i = 1 - \frac{r}{q}$$

where n is the total number of images within the result set that the tag t appears, r is the rank of the image i, and q is the total number of images within the ranked result set.

7.2.4 Tag Selection Based on Ranked Term Frequency

Using the Flickr interface, when users are prompted to create tags to describe the content of an image, it could be assumed they they will enter the tags that they deem most relevant to the image in descending order. This order is preserved within the data, and therefore could be considered as a ranked list. It is possible that these tags could be heterogeneous and only relevant through the users interpretation, making the ranking redundant for a single image.

It is logical to assume, however, that if there is a high level of correlation between high ranking tags over a result set of images, that these correlated tags could be deemed most relevant semantically. An evaluation was carried out across all top ranking tags within each result result set. Similarly to the ranked image approach evaluated, two different ranking schemes were utilised. The first ranking scheme places a large weight on tags that were ranked near the top of the lists. Tags that are ranked at the lower ends of the list are assigned a weight so low that they are effectively disregarded. This ranking scheme can be formulated as:

$$Score(t_j) = tf_j \times \sum_{i}^{n} w_j \quad \text{where} \quad w_j = 1 - \frac{1}{r}$$

where n is the total number of images within a result set that the tag t_j appears in and r is the rank of the tag t_j in image i.

The second ranking approach placed a more balanced weight distribution across all tag ranking positions. The variation in weights between top ranking and lower ranking tags is smaller than in the first ranking metric. This second approach is formally defined as:

$$Score(t_j) = tf_j \times \sum_{i}^{n} w_j \quad \text{where} \quad w_j = 1 - \frac{r}{q}$$

where r is the rank of the tag t_j in an image i, and q is the total number of tags retrieved for image i.

Tags were then ranked in a descending order based on $Score(t_j)$. The top ranked k tags were then chosen as the most representative tags for the retrieved image result set.

7.2.5 Tag Selection Based on Geographical Distribution

One approach that might help to solve tag selection issues using community data is the use of geographical information. Combining the geographical and textual based metadata that accompanies each image within the training corpus, should improve tag selection precision, as not only does a geo-tag have a semantic relationship with an image, it also has a semantic relationship with the associated textual metadata.

By calculating the spatial distribution of a tag throughout the whole corpus, it is hypothesised that it is possible to predict a relevant tag with a higher probability. A tag with a geographical distribution based over a small geographical area is more likely to be describing a landmark within that area, rather than a tag with a citywide geographical distribution. There are some exceptions to this rule, however, as it has been empirically noted that many Flickr users tag all of the images that they upload in a batch session with the same set of tags.

To indicate the geographically diverse distribution of each tag, a metric calculating the standard deviation was utilised. The standard deviation is a second order statistic that represents the amount of variation from the mean in a set of values. It is formally calculated using the following formula:

$$dev_i = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(x_i - \bar{x})^2}$$

where x_i is the geographical location for an ith instance of a tag and $\bar{x}$ is the mean geographical location of the tag. All standard deviation values are normalised in the range 0 - 1.

The actual score calculated for each tag is a combination of the tag frequency within the image result set and the geographical variation of the tag. This can be formally defined as:

$$score_i = tf_i \times (1 - dev_i)$$

It was found from experimentation that using a weighted value for tf_i performed with more precision. Based on this, two weights were evaluated:

$$score_i = w(tf_i) \times (1 - dev_i)$$

where w is equal to 2 and 4.

7.3 Tag Selection Evaluation

7.3.1 Introduction

To evaluate tag selection approaches, a benchmark selection of tags representing a number of ranked lists of images was created. The ranked results of 100 test image queries utilising the hybrid approach described in section 6.5 were collected. Tags associated with each image out of each of these ranked results were analysed manually. This benchmark consisted of a total of 602 images with an average of just under 6 tags per image, resulting in a total of 3444 tags. Each tag was deemed semantically relevant or irrelevant to the query image. This relevance was calculated based on a similar methodology to that described in section 3.3.2, which was a method to classify tag relevance into 1 of 5 categories. In this section, the 5 categories of semantic relevance in Chapter 3 were quantised into binary relevance scores, i.e. relevant or non-relevant:

- **Relevant - 1** A score of one is given to a tag that contains a high level semantic description. It must contain the name of the main landmark or surrounding geographical area (localised, not on a city wide scale), such as 'Notre Dame Cathedral' or 'Place de la Concorde'. A tag was also deemed relevant if it contains a mid-level semantic description of the content within an image. For example, if a tag describes the type of landmark or location depicted, or describes some additional information describing the part of a landmark that is photographed, it is deemed relevant. Some examples are: 'Cathedral', 'Facade' or 'Fountain'.

- **Non-Relevant - 0** A classification of non relevant is given to a tag that contains temporal information or a low-level semantic description of an image. Examples of a low-level semantic tag might be 'outdoor', 'sky', 'night', 'river' or 'park'. The tags 'Paris' and 'France' would also be deemed non-relevant, as the entire corpus is located within these locations. A non-relevant score is also given to a tag with very little or no relevance to the content contained within an image. For example, tags that contain vague geographical descriptions such as 'Europe','city' or 'continent' provide little discrimination value. Common heterogenous tags were also deemed irrelevant, including 'vacation', 'honeymoon' and 'trip'.

Each evaluated approach analysed different numbers of top ranked images k, where 1,2,3,4 and 5 was assigned to k. Four different evaluation metrics were utilised:

- **Precision** Precision is defined as the number of relevant tags selected for each test image divided by the total number of tags selected for that image.

- **Recall** Recall is defined in this task as the number of images where at least 1 relevant tag was selected.

- **F-Score** The F-Score or F-Measure is defined as the harmonic mean between precision and recall.

It is believed that for this task a balanced performance between precision and recall is desired. While the relevance of each selected tag is important, it is equally important to select at least one relevant tag for as many images as possible. Based on this, the F-Score metric is seen as the most important in the evaluation stage. A graph displaying the overall F-Score measures for each approach is displayed in Figure 7.2.

7.3.2 Term Frequency Selection

Two separate evaluations were carried out to analyse the effectiveness of frequency based approaches to tag selection. The first approach utilised the method described in section 7.2.1 using all the tags within each result set the results of which can be seen in Table 7.1. From informal empirical inspection, it seemed that the two terms 'paris' and 'france' had a disproportionate number of occurrences across all result sets, and therefore were repeatedly ranked as the top two tags for many of the query images. To account for this, a second approach was evaluated where all occurrences of these two tags were removed from the dataset. This is roughly analogous to stop word removal in document retrieval.

With the two terms removed, the results of the tag selection process improved dramatically. It would seem that the two removed tags were being undesirably selected for a large percentage of images analysed (approximately 27% where $k = 1$). Based on the results in Table 7.2, this was repeated for all subsequent evaluations.

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.24	.225	.28	.31	.34
Recall	.24	.41	.64	.82	.90
F-Score	.24	.29	.38	.44	.49

Table 7.1: Tag Selection - Tag Frequency Scheme

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.51	.49	.506	.51	.50
Recall	.51	.74	.88	.92	.95
F-Score	.51	.58	.63	.65	.65

Table 7.2: Tag Selection - Tag Frequency Scheme ('paris' and 'france' omitted)

7.3.3 TF-IDF

The results of the tf-idf evaluations can be seen in Table 7.3. From these results, it can be seen that the tf-idf method does not perform as well as the tag frequency approach evaluated in Table 7.2 in terms of precision or recall. The 2 approaches based on geographical distributions and tag rankings, evaluated in Tables 7.6, 7.7, 7.8 and 7.9, outperform the tf-idf scheme. It is assumed that this is due to the nature of the corpus.

In this corpus, there may be a high distribution of images representing a single landmark, and therefore a high distribution of tags describing the same landmark. This, in turn, significantly affects the precision of the tf-idf approach, as many semantically relevant tags would be incorrectly disregarded based on their idf scores. It would seem that although the tf-idf scoring metric is widely used in the literature for selecting representative landmark tags from unstructured sets of Flickr data, it is not optimal due to large distributions of relevant tags for commonly photographed landmarks throughout the corpus. It must be noted however, that for a different global based tag selection problem, the tf-idf approach might perform differently.

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.43	.44	.46	.45	.43
Recall	.43	.63	.78	.85	.87
F-Score	.43	.51	.57	.58	.57

Table 7.3: Tag Selection - tf*idf

7.3.4 Image Ranking Schemes

In this section, two tag selection schemes based on image similarity rankings are evaluated. Results of this evaluation can be seen in Tables 7.4 and 7.5. From these results it can be seen that the ranking of an image based on visual similarity within the result set does not necessarily correlate with tag relevance. Image ranking performed poorly in comparison to the schemes based on geographical variations, tag ranking, and term frequency when measuring precision. When analysing recall, however, the scheme performs quite well.

One reason for the poor performance of the image ranking schemes is that each image in the result set is likely to be visually similar based on the precision of the recognition framework (shown in Table 6.6), and when there is only a negligible difference in visual similarity within the result set there is unlikely to be a large semantic difference in the associated tags. Overall the calculated F-Score measure was below all of the other schemes.

7.3.5 Tag Ranking Schemes

In this section, the tag selection scheme based on tag rankings is evaluated. Results of this evaluation are presented in Tables 7.6 and 7.7. Two ranking schemes were analysed, one based on large weights assigned to high ranking tags and the

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.35	.39	.43	.42	.41
Recall	.35	.62	.79	.85	.89
F-Score	.35	.47	.55	.56	.56

Table 7.4: Tag Selection - Image ranking Scheme (weighting = $\frac{1}{r}$)

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.34	.35	.39	.37	.33
Recall	.34	.56	.74	.79	.82
F-Score	.34	.43	.51	.50	.47

Table 7.5: Tag Selection - Image Ranking Scheme (weighting = $1 - \frac{r}{q}$)

second based on a more evenly distributed weighting system. Interestingly, the first ranking scheme performed very poorly which would suggest that the top ranking tag is not necessarily the most semantically relevant. With a more evenly distributed weighting system, however, there is a significant improvement. This would suggest that although the actual top or top 2 ranking tags are not necessarily the most relevant, users would still assign relevant tags in the top portion of a ranked tag list.

7.3.6 Geographical Distribution Ranking Schemes

In this section, two selection schemes based on geographical distribution are evaluated. Results of this evaluation are presented in Tables 7.8 and 7.9. The geographical based scheme outperformed all of the other schemes evaluated. A weighting value of 2 for tf_i produced the highest F-Score results of all approaches. It would seem that tags that are limited to a specific geographical region have a higher probability of relevance to a specific landmark. It must also be assumed

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.22	.34	.38	.38	.41
Recall	.22	.59	.76	.85	.91
F-Score	.22	.43	.50	.52	.56

Table 7.6: Tag Selection - Tag ranking Scheme (weighting = $\frac{1}{r}$)

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.58	.60	.58	.54	.52
Recall	.58	.84	.91	.97	.97
F-Score	.58	.70	.70	.69	.67

Table 7.7: Tag Selection - Tag Ranking Scheme (weighting = $1 - \frac{r}{q}$)

that more generic tags that would still be considered relevant such as 'church' or 'statue' would have a high level of geographical variation due to their many instances within a metropolitan area. Therefore, the selected tags using this approach are more likely to contain the actual name of the landmark depicted, which is a desirable attribute.

When assigned a weighting score of 4 to tf_i, the results were somewhat erratic. If a value of 5 is assigned to k, the precision score drops by a significant percentage. It is assumed that this weighting measure is less stable than a weighting score of 2, therefore it is disregarded and the optimal weighting score is assigned to 2.

Results of the optimal weighting score enable the annotation of a test image with a relevance precision in the worst case of over 50%. These results are illustrated in Figure 7.2.

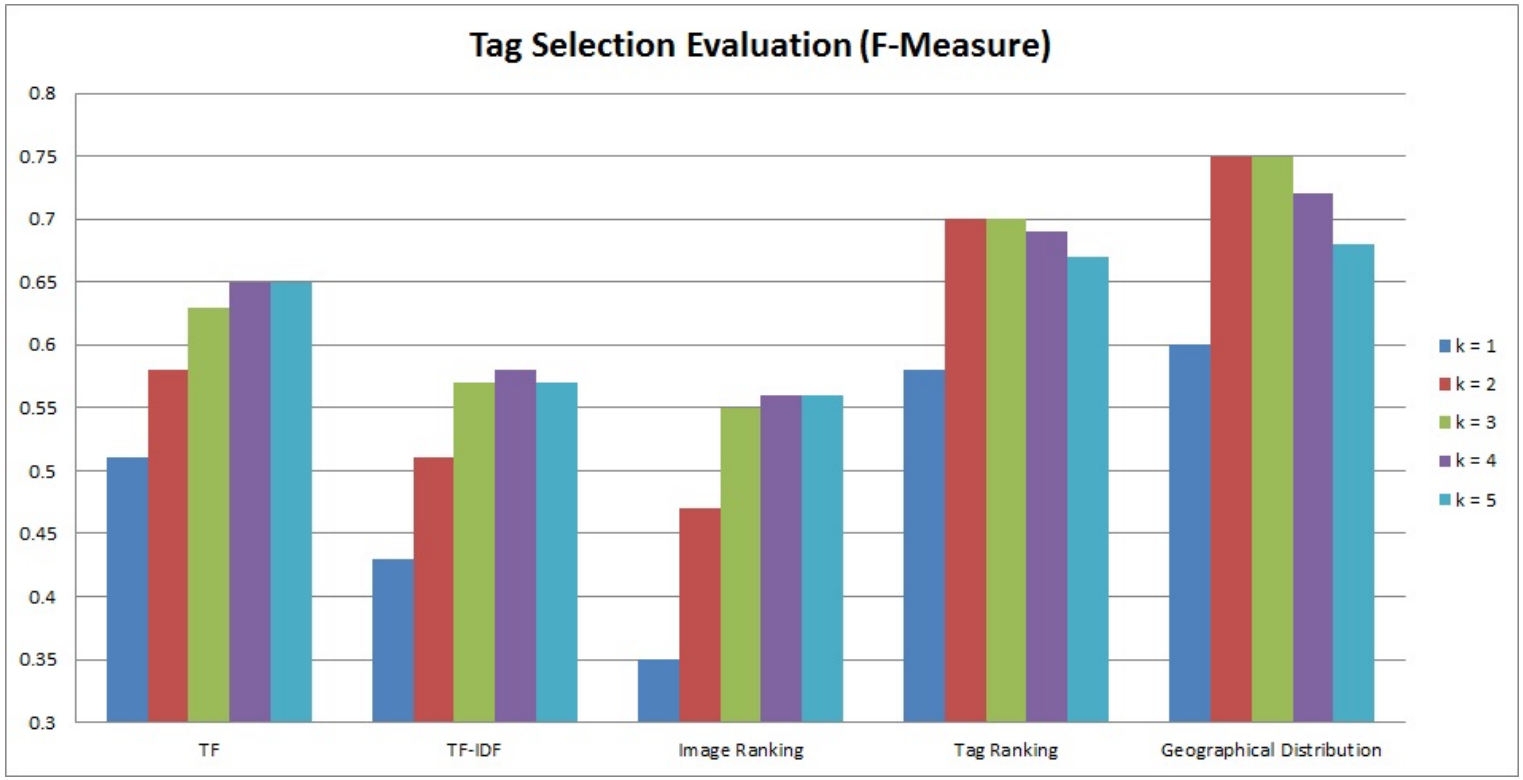

Figure 7.2: A chart comparing the f-score results output from each of the attempted tag selection schemes (optimal weighting values). Included are the results for 5 values for k where k is the number of top ranking images selected to represent a test image

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.60	.63	.62	.58	.53
Recall	.60	.88	.97	.98	.98
F-Score	.60	.73	.75	.72	.68

Table 7.8: Tag Selection - Geographical Distribution Scheme ($score_i = 2(tf_i) \times (1 - dev_i)$)

Number of Ranked Tags	k =1	k =2	k =3	k =4	k =5
Precision	.57	.60	.60	.57	.41
Recall	.57	.90	.96	.98	.98
F-Score	.57	.72	.73	.72	.53

Table 7.9: Tag Selection - Geographical Distribution Scheme ($score_i = 4(tf_i) \times (1 - dev_i)$)

7.4 Conclusions

In this chapter, several methods were proposed to garner semantic knowledge about a test image from a collection of relevant and non-relevant textual tags. The only structure that was available within the data was the knowledge that for each set of tags retrieved from an image, the order in which a user had entered those tags was preserved. As shown in Chapter 3, the majority of textual tags associated with this community data is heterogeneous, subjective, and bears minimal semantic relevance from an information retrieval perspective to the content of an image. In this chapter, an approach was proposed and evaluated that provides the means to extract semantic information with a high degree of precision.

The aim of this chapter was to propose alternative approaches to the tf-idf metric for scoring the relevance of Flickr tags within a visually similar result set. From the results of this evaluation, it can be seen that three of the four proposed approaches outperform the tf-idf method that has widely been used in similar tasks [Ahern et al., 2007] [Kennedy and Naaman, 2008].

From the results in Table 7.8, it is evident that the tag selection scheme based on geographical distributions performed with the most desirable level of precision and recall out of all evaluated approaches. When utilising a value of 5 for k, where k is the number of selected tags to annotate a test image, 98% of the test images had at least one relevant tag assigned to them. Additionally, when using this value for k there is an average precision score of over 50% which indicates that for every test image, there is on average 2.5 relevant tags assigned to it.

It must be noted that image title information was not taken into account in this work. From an informal empirical inspection, it was evident that a large number of image titles were vague and irrelevant. It was thought that the addition of this metadata would have a negative effect on many of the tag selection schemes and therefore it was disregarded.

It must also be noted that the proposed approaches to solve the problem outlined in this chapter do not take multi-lingual text into account, and the process could be further improved by utilising an established machine translation method. Additionally, issues exist with synonyms, which could be improved by giving relevant tags higher scores regardless of the metric if counted in tag frequency statistics for each of their synonyms.

In other similar work [Ahern et al., 2007], an additional metric is proposed to improve the tf-idf ranking based on user frequencies. Ahern et al. logically propose that a tag with a higher level of user frequency is less likely to be heterogeneous and more likely to have a high level semantic relevance to a specific region. In this work, the number of images retrieved within a result set is quite small (whereas Ahern et al. calculated this metric based on global distributions), and the probability of having an identical user within each of these result sets is minimal. Therefore no experimentation utilising user frequencies was conducted.

Chapter 8

Conclusion and Suggested Extensions

This thesis has focused on the creation and evaluation of a framework that allows for the automated recognition and annotation of landmarks within images. This framework provides an efficient method to carry out recognition in real time, where efficiency is measured in terms of recognition times for a single test image, and memory requirements for the framework. A full pipeline system to evaluate this framework was implemented, and each stage of the pipeline was evaluated throughout the course of this thesis.

8.1 Hypotheses

In Chapter 1, two main hypotheses were proposed:

- **Hypothesis 1.** *It is hypothesised that by structuring image data into semantically and visually related groups, that it would be possible to create a memory efficient framework based on machine learning algorithms to accurately classify commonly photographed landmarks within geo-tagged image corpora in real-time*

- **Hypothesis 2.** *It is hypothesised that by combining a machine learning based method with a commonly used tree indexing based approach that it is possible to improve upon existing methods to classify landmarks within digital images in a memory efficient manner*

The main hypothesis was that it would be possible to train a machine learning classification model with structured image data, to successfully recognise a commonly photographed landmark within a digital image. In this thesis, a framework was implemented and extensive experimentation was conducted to test this hypothesis.

The evaluation stage described in Chapter 5, provides evidence supporting this hypothesis. Experiments carried out reveal that a precision score of over .87 can be achieved, if analysing the top 3 ranking images. This is deemed to be an acceptable score given the size of the dataset. Additionally, it has been shown that this process can be carried out without the need for specialised hardware or additional memory on a standard 32-bit desktop computer. Results from this evaluation section show that the processing time required to achieve this recognition is just 3508 milliseconds, which includes the time required to extract image features from a query image. It is believed that this is an acceptable time frame to allow for interactive or real time recognition. From the threshold outlined in [Hoxmeier et al., 2000], it would seem that this timeframe would fit comfortably into the tolerable waiting time suggested for users of complex computing tasks.

The main advantages of using an approach based on machine learning methods is that the memory requirements are small and the time required to process a query image, allows for this process to be carried out in real-time. Many of the previously suggested approaches to landmark recognition rely on the use of an indexing structure, which is required to be loaded into heap memory. This places a restriction on the maximum number of images within a training corpus. It is estimated in [Philbin et al., 2007] that the inverted index structure for

a corpus of 105,000 images required 1GB of heap memory. This suggests that utilising this approach on a standard 32-bit machine would create a limitation on a corpus size to approximately 400,000. Other commonly used methods based on locality sensitive hashing and non-static vocabulary trees would require an even larger memory footprint. The work outlined in Chapter 5, improves on these techniques as it enables for the accurate classification of landmark images within a corpus that is only limited by hard disk space. The average memory footprint required for each multi-class classification model trained in this work was just under 4MB. Although workaround approaches have been suggested to account for this heap memory restriction, such as those based on using large numbers of parallel machines and 'forests' of vocabulary trees, these rely on expensive hardware, whereas the framework proposed in this work, can be run on a 32-bit desktop machine with 2GB of heap memory.

Another advantage of the machine learning approach is that, as the scale of the training corpus grows, the framework performs more accurately and requires less processing time. As new images are added to the training corpus, models become more robust, due to the additional information provided. Also, new models can be created, whereas previously, there was insufficient data. This provides a significant advantage over alternative approaches where their classification performance will decrease as the size of the corpus increases.

This novel approach to classifying landmarks based on the use of classification models improves on other similar techniques proposed in the literature [Li et al., 2009b] by structuring sets of models based on geographical information. Li et al. demonstrated that there was a significant improvement in using small number of classes in multi-class SVM models for landmark classification tasks. They have shown an increase of almost 200% in terms of classification accuracy between a 10-class classification model and a 500-class model. The framework outlined in this thesis structures data based on spatial information, and therefore,

ensures the number of classes per model can remain low. This is a significant improvement over work carried out in [Li et al., 2009b] as it allows for the classification of a larger number of landmarks while retaining a higher classification accuracy.

As part of this study, an extensive set of experiments were conducted to ascertain optimal parameters when classifying using machine learning techniques. A wide range of established computer vision features were evaluated. Some of the more advanced features that have been successfully used for image retrieval tasks performed quite poorly. Spatial pyramid based features, which provide additional geometrical information to a VBOW feature, have been shown to outperform VBOW features in many image matching and retrieval tasks. However, they achieved a precision and image recall score far below the optimal feature. It is assumed that this poor performance is down to feature vector length and a phenomenon in machine learning called 'the curse of dimensionality' [Pavlenko, 2003]. Overall the highest performing feature evaluated in terms of precision and image recall was the standard hard assignment, VBOW feature with a vocabulary size of 4096.

Results from the evaluation section in Chapter 6 demonstrating the use of the hybrid approach provide support for the second hypothesis. A commonly utilised method for indexing local image features, is a tree structure called a vocabulary tree. This tree structure provides a method to approximate nearest neighbour matches of interest point features. This hybrid framework has been shown to achieve a high precision score when evaluating the top ranking images retrieved. It has been shown that the hybrid approach outperforms a brute force matching method by a factor of approximately 100 in terms of required recognition time, while sacrificing a precision score of just 3.8%.

It has also been demonstrated that when combining an SVM approach with a vocabulary tree, that the required processing time is significantly reduced over

using just the vocabulary tree alone. This hybrid approach also has been shown to provide a more desirable image recall(relevant) score (recognising over 5% percent more test images), while sacrificing only a minimal decrease in precision (0.4%).

In Chapter 1, several secondary research objectives were defined. These objectives included:

- *Analysing the accuracy of community contributed metadata for the purposes of image matching and classification*
- *Propose approaches to automatically annotate query images by selecting subsets of semantically relevant tags from larger sets of noisy metadata.*

In Chapter 3, a large scale analysis of metadata that accompanies Flickr imagery was carried out. The main aim behind this analysis was to discern how much noise existed within the Flickr metadata, specifically how accurately the images were geo-tagged and the relevance of the contextual tag information. The outcome of this analysis showed that the geographical information that accompanies each image within the corpus was quite accurate. Over 80% of the geo-tags examined had an accuracy to within 200 metres. This analysis had enabled optimal parameters to be selected for spatial based search space pruning in the clustering and classification processes utilised in this work.

This analysis will also be useful for the wider research community as it improves upon previous work in the field. Many of the previous approaches to estimating image geo-tag accuracies concentrated solely on geographical data and thus were approximations with a potential large margin for error. Based on local knowledge and manual inspection, the analysis carried out in Chapter 3 provides a more accurate measurement of geo-tag accuracies.

Another objective of this thesis is to automatically provide a textual annotation for a query image based on a subset of all retrieved text tags. It is a challenging

problem to extrapolate semantically relevant tags from a large subset of noisy data. Several metrics to solve this problem were proposed in this work. Empirical evidence presented in Chapter 7, provides support for this research objective.

Many of the suggested approaches in the literature are based upon a variation of the tf-idf algorithm. This work improves upon this technique by analysing additional contextual information that is available within the metadata. It has been shown that three of the proposed approaches outperform the tf-idf methodologies.

8.2 Summary

This work proposes an end to end framework was proposed to solve the problem of automatically classifying landmarks within geographical image collections and automatically provided a relevant and accurate caption for a landmark image. The key contributions of this work are:

1. It is possible to automatically recognise specific landmarks within digital imagery, using a machine learning approach, with a high degree of precision by structuring data based on visual and geographical similarity. Additionally, is possible to conduct out this recognition procedure in real time in a memory efficient manner.

2. A thorough investigation was carried out to ascertain optimal parameters to be utilised when adopting classification models for the purposes of landmark recognition.

3. By combining two disjoint methods for landmark classification (SVM and Vocabulary Tree) using community datasets, a significant gain can be made in terms of required classification time and image recall.

4. A detailed analysis of community datasets was provided. The accuracy of their associated metadata was analysed and evaluated for use in image

matching an annotation tasks. The results of this analysis demonstrated that a large percentage of geo-tags were accurate to within 200 metres.

5. Several approaches to automatically rank a tags semantic relevance to an image were proposed and evaluated. Three of these proposed approaches outperform the most commonly used approach in the literature.

8.3 Future Work

There are several potential avenues for future research based on the outcomes of this thesis.

Based on the positive results from the evaluation in Chapter 6, it would seem logical to evaluate this work using much larger image corpora. At present, there are over 100,000,000 geo-tagged images stored in the Flickr repository. While this work examined a small subset of that collection, it is believed that the framework described in this thesis could be readily scaled up to process an image corpora of that scale. Geographical filtering would ensure that the discrimination values and real time classification times are retained. Using small, spatially organised classification models, the heap memory footprint would remain minimal. Scaling up this framework to index a data collection of that scale would create an immensely powerful image classification tool. Due to large distribution of the Flickr data worldwide, users of the framework could capture an image of a landmark from anywhere on the planet, and in real time, have that image recognised, annotated, and uploaded to a social network or photo sharing repository of their choice.

Another research avenue is to pursue web based knowledge retrieval. There is a wealth of knowledge available online in commercial and community created repositories. If a collection of relevant text tags is selected, using the approaches outlined in Chapter 7, it will be possible to use these tags as query terms into internet based search engines. A wealth of contextual knowledge becomes acces-

sible, and it becomes possible to annotate query images with rich annotations. Using these annotations it would be possible to create a virtual tour guide for a user, using only a mobile phone device.

It is also intended to improve on the accuracy of the tag selection schemes outlined in Chapter 7, using community contributed textual repositories and more advanced text processing methods. One potential improvement to the tag selection process, described in Chapter 7, is to utilise the online lexical datasets such as 'WordNet' [WordNet, 1985]. WordNet groups sets of English words into collections of synonyms, called 'synsets'. In total, the dataset consists of over 150,000 English words, and would provide a valuable resource to measure co-occurrences between sets of textual tags. For example, from the synset associated with the word 'Church', it would be possible to associate the semantically similar words 'Chapel' and 'Cathedral'. It is intended to explore the use of this data in future work.

Appendix A

Mobile Based Landmark Recognition System

A.1 Introduction

In this section, some related work is described that was carried out based on the research presented in this thesis. To analyse how this framework performs in real world conditions, a mobile phone based landmark classification system was developed and evaluated. The system allows for a user to take a photograph and have it automatically captioned using the hybrid framework before uploading the annotated image to a online photo repository of the user's choice, such as Flickr [Flickr, 2004] or Facebook [Facebook, 2004].

A user evaluation was also carried out on this system, including an evaluation of the accuracy of the landmark recognition framework and a user evaluation of the relevance of the annotations created by the system. It must be noted that some of the work in this appendix was carried out by Daragh Byrne and Dr. Andrew Salway. Additionally, some of this work appeared in [Jones et al., 2010].

A.2 Mobile Landmark Classification

The goal of the mobile classification system is to analyse the effectiveness of the landmark recognition framework in a real world scenario, and to gather user feedback on the effectiveness and accuracy of the framework. This system provided some novel improvements upon the framework described in this thesis in that it made use of online resources to provide a detailed annotation of a query image. For example, given an image with GPS data of its location of capture, the system returns a semantically-rich annotation comprising of tags which both identify the landmark in the image (as described in Chapter 7), and a list of interesting facts about it, e.g. 'A view of the Eiffel Tower, which was built in 1889 for an international exhibition in Paris' or 'A view of Le Tour Eiffel. Le Tour Eiffel was built for the International Exhibition of Paris of 1889 commemorating the centenary of the French Revolution'.

The system exploits visual and textual web mining in combination with content-based image analysis and natural language processing. In the first stage, an input image is matched to a set of community contributed images (with keyword tags) on the basis of its GPS information and image classification techniques using the hybrid framework described in this work. The depicted landmark is inferred from the keyword tags associated with the matched set of images. The system then takes advantage of the ample information written about landmarks available on the web at large, to extract a fact about the landmark in the image.

A.3 Application

This system was created for use on Apple's smartphone, the 'iPhone 3GS'. Additionally, it may also be used on Apple's tablet computer, the 'iPad'. The iPhone has an integrated digital camera along with an integrated GPS receiver. The

integrated camera allows for the capture of digital imagery with a sensor size of 5 megapixels.

The application operates as follows: first the user selects a photo they want to process, either by taking a new image with the device's in-built camera or by selecting an existing image from the photo library. They are then asked to confirm that the location for the image is correct, after which the image and location data is passed to the middleware layer through a REST-based API. After the service completes the matching and annotation of the image, a response is returned to the device. The annotated image is then saved to a local data store and the application presents the results on-screen. The image, along with the automatically generated captions and tags, can then be uploaded to a number of social media sites including Flickr and Twitter through the results screen. An overview of the system architecture is presented in Figure A.1.

From a users perspective the mobile application is comprised of several interface screens:

1. Once a user takes an image, the system then analyses the GPS information on the device and provides a map based interface, centred on the users current location. The system then provides the user with an opportunity to amend the GPS information in case of an error, where they can pan and zoom to a correct location. This is illustrated in Figure A.2.

2. The system then sends the image across to a server running the landmark recognition framework. The server carries out image recognition, image annotation and retrieves information from online web services describing the landmark depicted.

3. After processing, the first screen on the device that is presented to the user consists of the all annotations created to describe the image. This is illustrated in Figure A.3.

4. The second screen displays all matched images from the corpus so that the user can confirm that their image was correctly matched. This is illustrated in Figure A.4.

5. The final screen that is available to a user displays social media websites, where the user can upload the annotated image to. This screen is depicted in Figure A.5.

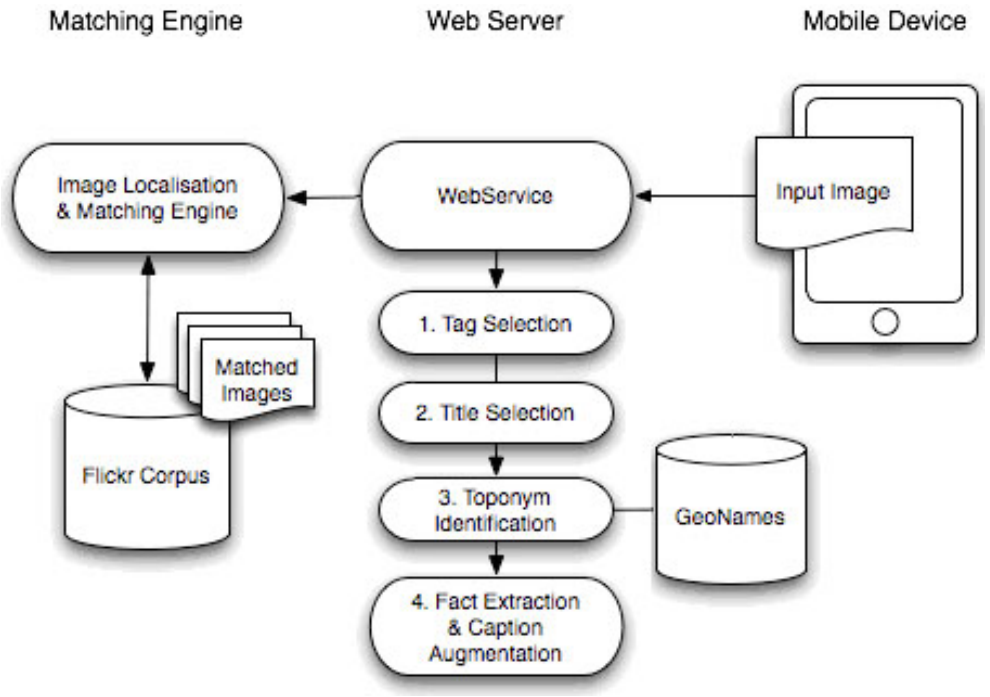

Figure A.1: An illustration demonstrating how all components of the system fit together and interact with each other.

A.4 System Pipeline

The system recognition pipeline consisted of four main stages:

1. Landmark Recognition

2. Tag Selection

3. Toponym Identification

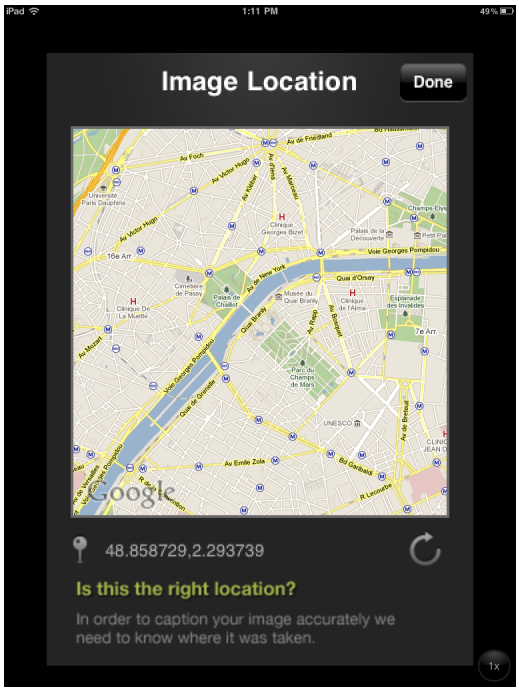

Figure A.2: Once an image has been taken by the system, the application will automatically provide a map based interface, centred on the current GPS coordinates. A user then has the option to refine the geo-tags to account for inaccurate GPS coordinates. The user can pan and zoom to a more accurate location using this map interface.

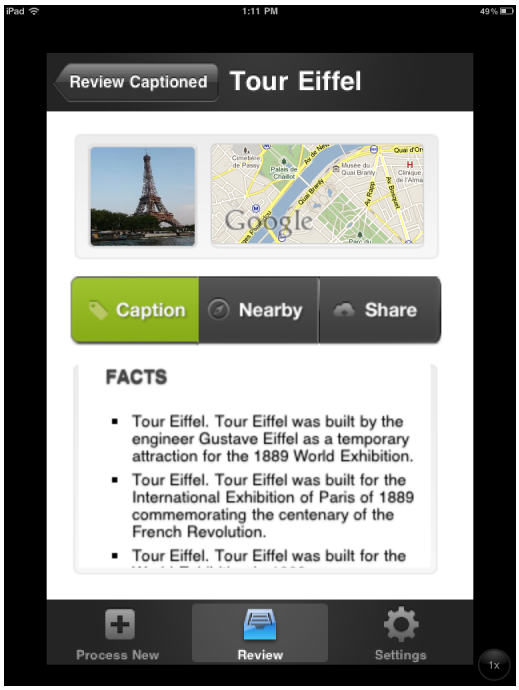

Figure A.3: The application allows a user to browse through the selected captions and historical facts describing the landmark that they depicted.

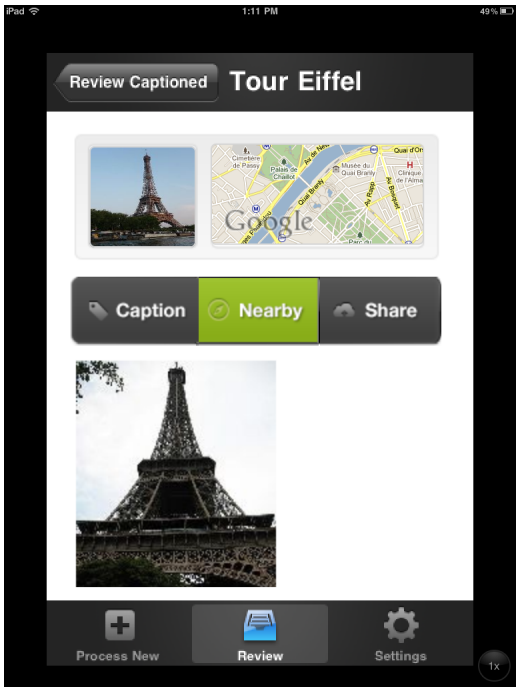

Figure A.4: The application allows a user to browse through the images from the corpus that their image was matched with. This stage allows for a user to confirm that indeed their image was correctly classified before they upload the photo to an online repository.

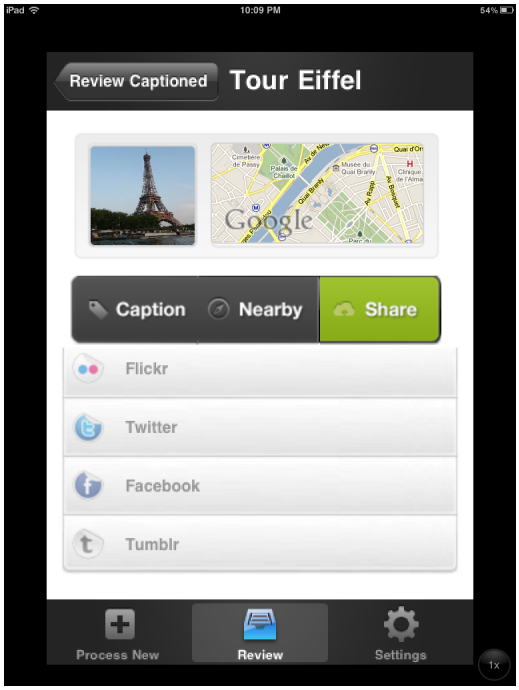

Figure A.5: The application provides the user with an option as to which online image repository they would like to upload their captioned image to.

4. Fact Extraction and Title Augmentation

A.4.1 Landmark Recognition

The landmark recognition framework used in this mobile system is described in Chapter 6 of this thesis. It consisted of a hybrid based approach to landmark recognition. This hybrid approach consisted of a collection of spatially organised SVM classification models, which were used to classify commonly photographed images of landmarks, and a hierarchical based approach, which was used to classify uncommonly photographed images of landmarks. This hybrid approach is described in more detail in section 6.5.

A.4.2 Tag Selection

Each matched image returned by the system has associated contextual information which includes a set of tags. It was desired to assign the most relevant tags to the query image. To achieve this, the set of returned images were examined and an attempt to reach a consensus on appropriate tags to be applied to the input image was made. The community-contributed tags were leveraged within the result set to achieve this. A hashmap of tags is created from the available tag set, by iterating through the matched images and the tags they contain and adding them to the set. This yields a set of distinct candidate tags. Each tag within the set is then given a weighting which corresponds to both the number of occurrences within the result set and the order of the appearances within the ranked results list (similarly to the image ranking approach outlined in section 7.3.4). To do this, each tag is assigned an initial weight of zero and a weighting component, a real number in the range of zero to one, is provided to initialize. The set of image results is then iterated over, and each tag for the result is examined. As a tag is encountered, that tags score is incremented by the initial weighting measure

raised to the exponent of its index within the result set. This gives precedence to the tags within the top ranked elements in the result set i.e. tags encountered in the highest scoring match are given greater weight than those from images further down the result set. Once each tag has been ranked, a thresholding step is then applied to prune the candidate set to a representative set. In order to perform this step, the percentage of the total tag set that it should be constrained to must be defined. Setting the bound to be a percentage value is not ideal in all cases, however, as in some over several hundred tags may occur in result sets and this would result in potentially large numbers of tags being applied to the image. Within increased numbers, the possibilities for more noisy output are introduced and heuristically the tag set should be a relatively small number of between 6-10 tags. As such the thresholding approach, in cases of large input tag sets may lower the target size, e.g. to around 10 items. Once a target size has been set, the scored tags are ordered by importance and the item on the boundary of the target set is identified. To further prevent noisy sets a heuristic is applied at this point by examining the number of items above the threshold score and the number at the threshold point. If more than 60% of the target set above the threshold and a further 30% or more of the tags lies on the threshold, only the items above the threshold score will be included. This prevents cases where a large number of items may lie on the threshold being accepted into the candidate set. This thresholding approach using heuristic selection methods is designed to limit the number of tags selected to a small but representative set which can be used to annotate the target image.

The selection of appropriate tags for the target image is extremely important within the workflow not only in terms of selecting tags for the image, but since these tags are the source of input to web-based augmentation stage. In order to evaluate the accuracy of the tagging phase. 150 test images were groundtruthed in order to validate tag selection. The tag results returned by the image matching

service were formed into a union pool set. An annotator manually judged each tag and made a binary classification of it relevance. A tag was determined to be relevant if it described the landmark featured in the image. As such, while potentially useful in describing the image or its composition, associated tags were not deemed relevant. Thus details such descriptions of camera types, related tags such as weather or lighting, or information on the year or events and activities were deemed non-relevant, since the emphasis in the groundtruthing was placed on tags most useful for use in the augmentation stage. On average 5.95 tags were deemed relevant per image, while an image had a average of 42.76 tags taken from 7.58 matches from the corpus. With a groundtruth established, the tag weighting and thresholding approach as previously described was applied to the image matching results for each test image. The input parameters were varied from 0.5-0.95 for both weighting and threshold and all of the combinations iterated. The set of tags returned from each variation was compared against the groundtruth for that image. Precision and recall measures were calculated as outlined in [Jschke et al., 2009]. These were then averaged across all of the test images and the f1m measure calculated. Within the selection of tags there's a need to balance precision and the recall so that 'noisy' or heterogeneous tags are kept to a minimum while a maximum of the relevant tags are contained within the selected set. Applying a low threshold results in a more unconstrained and noisy set displaying high recall but lower precision. Conversely, a high threshold and weight, results in a more constrained set, which will displaying high precision, has lower overall recall. The best precision (0.623) was found to occur with a harsh threshold of 0.95 with a decrementing weight of 0.85. The worst precision (0.516), but best recall (0.761), was found to occur with a threshold of 0.5 with a weighting of 0.95. The poorest recall occurred almost opposite to the best precision score (0.651), with a threshold of 0.95 and weight of 0.8. Given that the best and worst scores for both precision and recall almost mirror each other, there is need to

carefully balance the tag selection performance for both. By exploring the various variations for the highest f1m (0.644), a threshold value of 0.85 was identified with a iteratively decreasing tag weighting of 0.95 to be optimal for tag selection. This results in an average of 6.92 tags being selected, which is very close to the desired number of tags as indicated by the annotation effort.

A.4.3 Toponym identification

A toponym is used to initiate the fact extraction and title augmentation step, and its accuracy is important to ensure the effectiveness of the fact extraction stage. The identification of an appropriate toponym for the images is reliant upon the outputs of the tag filtering and selection process as outlined previously. To investigate this, 150 test images were selected, image matching was performed and a set of tags filtered from the results selected. Nearby toponyms were looked up using GeoNames and a candidate selected based on the tag set in a manner as outlined in Section A.2.2. Each of the returned toponyms were then annotated into one of the following categories: Incorrect toponym identified; Vague or unspecific toponym identified, e.g. Paris, France; Toponym is related to the target but is incorrect, this included a landmark nearby or within the image but which was not the primary focus or featured landmark, e.g. the Champ de Mars returned in place of the Eiffel Tower; and finally a correctly identified toponym. In total 30 of the toponyms were incorrect, 16 were vague, 21 were incorrect but related and 83 were correct.

While 55.33% of the tested images returned a correctly identified toponym, a further 14% (vague and related categories) may be considered acceptable (totalling 69.33%). All of the vague cases were composed of a generic toponym of 'Paris', which return facts such as 'Paris is named after a Celtic tribe called the Parisii who lived on the island in the river', 'Paris is famous for its huge number of cafes and

brasseries' and 'Paris was made for lovers and lovers of life.' While these facts are not ideal, they are generic enough to be reasonably acceptable. Additionally those which are related often contained reference to the target landmark. For example, in the case where the Champ de Mars was identified in place of the Eiffel tower, the first returned fact is the following: 'Champ de Mars is a green area located in the middle of the Eiffel Tower and the Ecole Militare building'. To ascertain the reason for poor toponym selection the 30 cases were compared where an invalid toponym was selected and applied to an image against the 83 cases where it was successful. The invalid cases on average received only 3.03 results from the image matching step (min 1, max 11, median 2) while the successful toponym cases had an average of 8.94 results (min 1, max 86, median 4.5). The successful cases have higher number of results on average, often substantially higher, and can as a result reach greater consensus on the appropriate tags. This most likely positively affects the toponym selection step. Additionally, the invalid cases, had on average 12.77 distinct tags returned within the results set, of which 6.6 were selected as representative tags for the target image. With the successful cases, there were 26.76 distinct tags of which 6.06 were selected. Having a lower number of tags and less diversity in the tags may make it more difficult to filter and threshold the tag set successfully. In the case of the successful set it can be seen that the set of available tags is being more judiciously pruned to 22.6 of the original set in comparison to 51

A.4.4 Fact Extraction and Title Augmentation

In the next stage of processing the output of the image classification and toponym identification is used as input to a highly portable mechanism for the extraction of partially structured facts from information on toponyms available on the World Wide Web. A particular feature of this is that it exploits information redundancy

on the web, i.e. the fact that the same information about a landmark is available in many forms on the web. This method is described in detail in [Salway et al., 2010]. For a given landmark, a list of facts is returned in the form (Landmark, Cue, Text-Fragment), ranked according to a score which is intended to promote interesting and true facts. This fact structure makes it straightforward to combine it with an existing image title. Crucially, for this information extraction process, it is assumed that at least one key fact about a landmark will be expressed somewhere on the web in a simple form, so that its only necessary to work with a few simple linguistic structures and shallow language processing. The following sub-sections describe the fact extraction process.

Get Snippets from Search Engine: A series of queries is made to a web search engine (Yahoo's BOSS API [Yahoo, 1995]). Each query takes the form <"Landmark Cue">; where the use of double quotes indicates that only exact matches are wanted, i.e. text in which the given landmark and cue are adjacent. A set of cues is manually specified to capture some common and simple ways in which information about landmarks is expressed, e.g. 'is a', 'is famous for', 'is popular with', 'was built'.

Although around 40 cues were examined (including single / plural and present / past forms), a much smaller number are responsible for returning the majority of high ranking facts; in particular (and perhaps unsurprisingly) the generic "is" seems most productive. The query may also include a disambiguating term. For example, streets and buildings with the same name may occur in different towns, so a town name can be included in the query outside the double quotes, e.g. <"West Street is popular with" Bridport>. For each query, all the unique snippets returned up to a preconfigured maximum number are processed in the next step. Typically a snippet is a few lines of text from a webpage around the words that match the query, often broken in mid-sentence.

Shallow Chunk Snippets to Make Candidate Facts: The system is only only retrieving information about a given landmark that is expressed as "Landmark Cue ...", therefore, a simple extraction pattern can be used to obtain candidate facts from the retrieved snippets. The gist of the pattern is 'BOUNDARY LANDMARK CUE TEXT-FRAGMENT BOUNDARY', such that 'TEXT-FRAGMENT' captures the 'Text-Fragment' part of a fact. The details of the pattern are captured in a regular expression on a language-specific basis, e.g. to specify boundary words and punctuation, to allow optional words to appear inbetween LANDMARK and CUE, and to reorder the elements for non-SVO languages. A successful match of the pattern on a snippet leads to the generation of a candidate fact. For example, using extraction patterns the snippet text '...in London. Big Ben was named after Sir Benjamin Hall. ...' matches, giving the candidate fact (Big Ben, was named, after Sir Benjamin Hall) but 'The square next to Big Ben was named in 1848...' does not match.

Filter Candidate Facts: Four filters are used as a quality control to remove candidate facts that: contain potentially subjective words; end in words that would be ungrammatical; are under a length threshold; and that contain words that are all in capitals. Finally, facts are ranked so that it is more likely to get correct and interesting facts at the top. The overlap between candidate facts is exploited for the same Landmark-Cue pair to capture these notions to some extent. For each Landmark-Cue pair a keyword frequency list is generated by counting the occurrence of all words in the Text-Fragments for that pair, words in a stopword list are ignored. The score for each fact is then calculated by summing the Landmark-Cue frequencies of each word in the Text-Fragment, so that facts containing words that were common in other facts with the same Landmark-Cue will score highly. If shorter facts are wanted then the sum is divided by the word length of the Text-Fragment.

The sum score for a fact can become high in two ways: (i) there are many overlapping Text-Fragments for an Landmark-Cue pair, so there are some high word frequencies; and (ii) a fact contains more of these high frequency words than other facts. Thus, the method is designed to highly rank facts with the most appropriate Cue for the Landmark, and the best Text-Fragment for the Landmark-Cue pair. For an existing image title, e.g. "A view of the Eiffel Tower", then the top-ranked fact, e.g. 'Eiffel Tower, was built, in 1889 for an international exhibition in Paris', can be inserted in one of two ways: (i) as a new sentence - "A view of the Eiffel Tower. The Eiffel Tower was built in 1889..."; or (ii) as a subclause - "A view of the Eiffel Tower, which was built in 1889...".

A.5 User Evaluation

The previous sections have outlined detailed evaluation of each component of the overall system. In order to validate its overall performance, a user-evaluation was conducted to examine the overall perception of the system's end-to-end performance in captioning landmark images.

A.5.1 Participants

15 study participants (12 male, 3 female) took part in the evaluation. Participants were selected opportunistically and were expected to have some prior knowledge of Paris in order to perform the evaluation. All of the participants were staff or postgraduate students within the DCU School of Computing faculty and had good computing experience. None of the participants had previous exposure to the user interface, though some familiarity with the concepts and technologies could be expected. All users had some previous experience with interactive multimedia systems. No incentive to participation was provided.

A.5.2 Evaluation Method

For the evaluation, 16 test cases were selected from the test corpus of 1000 Parisian landmark images. Eight of these cases were popular and well-known landmarks which occurred regularly in the corpus. These include for example the Eiffel Tower and Notre Dame Cathedral. It was expected that for these popular landmarks, the system should perform well in all aspects of the captioning process and this should result in a good overall annotations being applied to the images. Additionally, eight less popular and more challenging cases were evaluated. These included less prevalent landmarks such as the College of the four nations and more difficult landmarks such as statues e.g. the Thinker by Rodin.

Participants were provided with the hardware, an iPhone, required to complete the evaluation. As the users had no previous experience with the interface, they were instructed on its used and asked to familiarize themselves with the search system for a short period prior to commencing the evaluation. Once familiar with the system, they were asked to complete a series of topics. The order of topics was organized to maximize the coverage of the topics. Nine participants completed 12 topics while the remaining 6 participants completed 6 topics. In total 144 captioning judgments were made. Users were allowed to complete their assigned topics at their convenience but were encouraged to do so without interruption.

A questionnaire was administered across various stages of the evaluation. Prior to commencing, background and demographic information, along with familiarity with similar systems was captured. After each topic, the users were asked to provide subjective ratings on the systems performance for that topic. Finally, after completing all assigned topics, the participants were asked to provide general feedback on their experiences with the user interface including the System Usability Scale (SUS) [Brooke, 1996].

A.5.3 Results and Discussion

Within this section, the outcomes of the end-to-end user evaluation are discussed. The findings are presented in terms of the system's usability and its efficacy in captioning landmark images.

System Usability

The participants were generally unfamiliar with the interface. This scored an average 1.53 on a 7-point Likert scale. Despite their unfamiliarity with the interface, they found it both very easy to use and very easy to learn. These components scored 5.46, and 6.23 respectively on a 7-point scale. The participants additionally provided qualitative feedback that supports its ease of use. When asked what they liked about the system participant 6 indicated that it was easy to use while participant 5 noted it being fast and simple. Participant 12 remarked upon the overall user inteface: user interface is fluid nice flow.

To further evidence the general usability of the system, it scored favourably with the SUS questionnaire. The user interface scored 79.375%. Within the SUS scale, the system scored very highly for its ease of learning. For question 7 (I would imagine that most people would learn to use this system very quickly) and question 10 (I needed to learn a lot of things before I could get going with this system) scored 4.333 and 1.166 on the 5-point Likert Scale respectively. The system was not found to be cumbersome, scoring 1.416 for Question 8 and the users felt very confident using the system (4.0, Question 9) even with limited familiarity and a short training time.

System Performance

Following the completion of the topics the users were asked to score the system generally in terms of its responsiveness and performance at captioning. This was

rated again on 7-point Likert scale and the system scored favourably on each criteria. The system was found to be very responsive (mean 5.46). The system was also found to perform fairly well for its overall captioning effectiveness (mean 4.77), and its performance at providing titles using the matched toponym and at providing suitable facts for the landmark was viewed as performing similarly well (4.77 and 4.85 respectively.) The systems performance at selecting and applying tags to the image was seen as slightly less effective scoring neitherly positive nor negatively with 4.08.

Two freeform questions probed what the participants liked and disliked about the system and this finds similar sentiment to the quantitative scores outlined above. Participant 2 noted the system to be really accurate and fast while participant 5 liked its overall responsiveness, commenting on its quick collection of facts and tags. The facts were viewed very favourably by the participants, most liking their inclusion and noting their utility. Participant 9 commented that the system had Good captions performance and that the quality of facts proposed was generally high. In particular, Participant 14 liked the facts provided by the system noting them to be pretty accurate for the most part and the most informative of all the data. The tags selected by the system were viewed somewhat less favourably and participant 14 describing them as not always relevant. The qualitative feedback and the lower perceived performance indicate the participants were very discerning about noisy or irrelevant tags.

The participants also completed a per-topic subjective rating of the systems perceived performance. They were asked following each topic to rate the systems performance at captioning that image on a 7 point Likert scale. A 55% confidence in the systems captioning performance was found (4.86 mean, 1.94 standard deviation.) The general perception of the system was it had fair performance overall. The per-topic performance is now explored in more detail. The full details

of the average topic performance and more detailed information on the test case topics can be found in Table n below.

Within the evaluation the topics were divided into two groups. It was composed of 8 cases relating to landmarks which were prevalent and popular, while a further 8 cases appeared far less regularly within the test and training set. As anticipated, with more exemplar images and more matched results, the popular cases performed far higher than their sparse counterparts. The popular cases rated a mean of 5.80 (or 68.7% confidence) with the sparse cases 3.953 (or 42.2% confidence). From this it is asserted that the more prevalent a landmark is within a corpus the more effective the system will be at captioning it. This is because much of the captioning process relies on gaining consensus from the community contributed annotations. In order to perform well there must be a sufficient number of results to disambiguate relevant and non-relevant tags and allow the system to gain consensus. Where there are low numbers of results it will be more difficult for the system to achieve this.

To further explore this, the topics are divided into three groups: those that returned more than ten results, those that returned between 10 and 1 result, and those that returned just one result. There was 5 cases that had more than 5 results (max 132, min 19, average 55.8). These cases because of their larger result set has much larger numbers of tags (mean 95.8 distinct tags, max 157, min 36) from which a relatively small constrained set were chosen (mean 7.8 selected, max 11, min 6). As the tags for this set were very judiciously selected and constrained to on average just 7.9% of the original tags, a strong consensus on the relevant tags is reached. The participants agreed and rated these cases with an average system captioning confidence of 74.1% (mean 6.19 on 7-point scale). The 8 cases with between 1 and 10 image results faired less favourably scoring just 52.6% confidence (mean score of 4.68). The difference in the results and tag set is marked. These images included on average 3.88 results (min 2, max 7) with on average

23.88 distinct tags (min 12, max 47) from which an average of 8.13 tags were selected (min 3, max 32). This represents a selected set of 34% of the original set and is far less constrained than the cases with larger results sets. Finally, three cases had just one result. These were scored very poorly with just 31% confidence in the captions output. All of these items had only one image match, and were reliant on that image for candidate tags making the likelihood of noise being introduced extremely probable.

From this it can be seen that the captioning performance is closely linked with the number of matched results. In cases of well matched images the system performs very well, with a perception of 75% captioning effectiveness. This is an extremely encouraging result.

List of Publications

- G. J. F. Jones, D. Byrne, M. Hughes, N. E. O'Connor and A. Salway. "Automatic Semantic Annotation of Landmark Images with Web Mining," 5th International Conference on Semantic and Digital Media Technologies (SAMT 2010) , December 2010.

- M. Hughes, G. J. F. Jones and N. E. O'Connor. "Investigation of Image Models for Landmark Classification," 4th International Workshop on Semantic Media Adaptation and Personalization, December 2009.

- M. Hughes,G. J. F. Jones and N. E. O'Connor. "A Social Framework for the Organisation and Automated Annotation of Personal Photo Collections," 3rd International Workshop on Semantic Media Adaptation and Personalization, December 2008.

- M. Hughes. "Determining Spatial Classification Models for Automated Landmark Identification," K-Space Jamboree Workshop, July 2008.

- A. Doherty, D. Byrne, A. F. Smeaton, Gareth J F Jones and M. Hughes. "Investigating Keyframe Selection Methods in the Novel Domain of Passively Captured Visual Lifelog," ACM International Conference on Image and Video Retrieval, July 2008.

- M. Hughes and G. J. F. Jones. "Analysing Image-Text Relations for Semantic Media Adaptation," VGV08 - Irish Graduate Student Symposium on Vision, Graphics and Visualisation, December 2008.

- M. Hughes, A. Salway, G. J. F. Jones and N. E. O'Connor. "Analysing Image-Text Relations for Semantic Media Adaptation and Personaliation," 2th International Workshop on Semantic Media Adaptation and Personalization, December 2008.

Bibliography

[Abbasi et al., 2009] Abbasi, R., Chernov, S., Nejdl, W., Paiu, R., and Staab, S. (2009). Exploiting flickr tags and groups for finding landmark photos. In *Advances in Information Retrieval*, pages 654–661.

[Ahern et al., 2007] Ahern, S., Naaman, M., Nair, R., and Yang, J. (2007). World explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. In *In Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 1–10. ACM Press.

[Alexa, 2010] Alexa (2010). Top 500 global websites. retrieved on 17th august 2010. `http://www.alexa.com/topsites`.

[Aly et al., 2009] Aly, M., Welinder, P., Munich, M., and Perona, P. (2009). Scaling object recognition: Benchmark of current state of the art techniques. *2009 IEEE 12th International Conference on Computer Vision Workshops ICCV Workshops*, pages 2117–2124.

[Annesley et al., 2005] Annesley, J., Orwell, J., and Renno, J.-P. (2005). Evaluation of mpeg7 color descriptors for visual surveillance retrieval. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 0:105–112.

[Ashley et al., 1995] Ashley, J., Flickner, M., Hafner, J., Lee, D., Niblack, W., and Petkovic, D. (1995). The query by image content (qbic) system. In *SIGMOD '95:*

Proceedings of the 1995 ACM SIGMOD international conference on Management of data, page 475, New York, NY, USA. ACM.

[Avrithis et al., 2010] Avrithis, Y., Kalantidis, Y., Tolias, G., and Spyrou, E. (2010). Retrieving landmark and non-landmark images from community photo collections. In *in Proceedings of ACM Multimedia (Full paper) (MM 2010)*, Firenze, Italy.

[Ayers and Boutell, 2007] Ayers, B. and Boutell, M. (2007). Home interior classification using sift keypoint histograms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 0:1–6.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and L., V. G. (2006). Surf: Speeded up robust features. *9th European Conference on Computer Vision*, pages 404–417.

[Bentley, 1975] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517.

[Birchfield and Rangarajan, 2005] Birchfield, S. T. and Rangarajan, S. (2005). Spatiograms versus histograms for region-based tracking. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 1158–1163, Washington, DC, USA. IEEE Computer Society.

[Borghesani et al., 2009] Borghesani, D., Grana, C., and Cucchiara, R. (2009). Color features performance comparison for image retrieval. In Foggia, P., Sansone, C., and Vento, M., editors, *Image Analysis and Processing ICIAP 2009*,

Lecture Notes in Computer Science, pages 902–910. Springer Berlin / Heidelberg.

[Bosch et al., 2006] Bosch, A., Zisserman, A., and Muoz, X. (2006). Scene classification via plsa. In *European Conference on Computer Vision*, pages 517–530.

[Bosch et al., 2008] Bosch, A., Zisserman, A., and Muoz, X. (2008). Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):712–727.

[Brooke, 1996] Brooke, J. (1996). SUS: A quick and dirty usability scale. In Jordan, P. W., Weerdmeester, B., Thomas, A., and Mclelland, I. L., editors, *Usability evaluation in industry*. Taylor and Francis, London.

[Brown et al., 2005] Brown, M., Szeliski, R., and Winder, S. (2005). Multi-image matching using multi-scale oriented patches. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 510–517, Washington, DC, USA. IEEE Computer Society.

[Cai et al., 2004] Cai, D., He, X., Li, Z., Ma, W.-Y., and Wen, J.-R. (2004). Hierarchical clustering of www image search results using visual, textual and link information. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 952–959.

[Canny, 1983] Canny, J. F. (1983). A variational approach to edge detection. In *AAAI*, pages 54–58.

[Chang and Lin, 2001] Chang, C. C. and Lin, C. J. (2001). *LIBSVM: a library for support vector machines*.

[Chapelle et al., 1999] Chapelle, O., Haffner, P., and Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055–1064.

[Chatzichristofis et al., 2009] Chatzichristofis, S. A., Boutalis, Y. S., and Lux, M. (2009). Img(rummager): An interactive content based image retrieval system. *Similarity Search and Applications, International Workshop on*, 0:151–153.

[Chum and Matas, 2008] Chum, O. and Matas, J. (2008). Web scale image clustering – large scale discovery of spatially related images. In *Technical Report CTU-CMP-2008-1*.

[Chum et al., 2003] Chum, O., Matas, J., and Kittler, J. (2003). Locally optimized ransac. *DAGM 2003 Proceedings of the 25th DAGM Symposium*, 2781:236–243.

[Conaire et al., 2007] Conaire, C., O'Connor, N., and Smeaton, A. (2007). An improved spatiogram similarity measure for robust object localisation. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–1069 –I–1072.

[Connaire et al., 2009] Connaire, C. O., Blighe, M., and O'Connor., N. (2009). Sensecam image localisation using hierarchical surf trees. In *MMM 2009 - 15th international Multimedia Modeling Conference*.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. In *Machine Learning*, pages 273–297.

[Crandall et al., 2009] Crandall, D. J., Backstrom, L., Huttenlocher, D., and Kleinberg, J. (2009). Mapping the world's photos. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893.

[Datar and Indyk, 2004] Datar, M. and Indyk, P. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *In SCG 04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM Press.

[Datta et al., 2006] Datta, R., Joshi, D., Li, J., James, and Wang, Z. (2006). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 39:2007.

[Daugman, 1993] Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.

[Digg, 2004] Digg (2004). Website address. `http://www.digg.com`.

[Dwyer, 1987] Dwyer, R. A. (1987). A faster divide-and-conquer algorithm for constructing delaunay triangulations. *Algorithmica*, 2:137–151.

[Facebook, 2004] Facebook (2004). Website address. `http://www.facebook.com`.

[Fan et al., 2006] Fan, Q., Barnard, K., Amir, A., Efrat, A., and Lin, M. (2006). Matching slides to presentation videos using sift and scene background matching. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, MIR '06, pages 239–248, New York, NY, USA.

[Fei-Fei et al., 2004] Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, page 178.

[Fischler and Bolles, 1987] Fischler, M. A. and Bolles, R. C. (1987). Readings in computer vision: issues, problems, principles, and paradigms. chapter Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, pages 726–740. San Francisco, CA, USA.

[Flickr, 2004] Flickr (2004). Website address. `http://www.flickr.com`.

[Foo et al., 2007] Foo, J. J., Zobel, J., and Sinha, R. (2007). Clustering near-duplicate images in large collections. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, MIR '07, pages 21–30, New York, NY, USA. ACM.

[FourSqaure, 2009] FourSqaure (2009). Website address. `http://www.foursquare.com`.

[Frahm et al., 2010] Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., and Pollefeys, M. (2010). Building rome on a cloudless day. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg. Springer-Verlag.

[Gagaudakis et al., 2000] Gagaudakis, G., Rosin, P., and Chen, C. (2000). Using cbir and pathfinder networks for image database visualisation. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 1052 –1055 vol.1.

[Gan et al., 2008] Gan, Q., Attenberg, J., Markowetz, A., and Suel, T. (2008). Analysis of geographic queries in a search engine log. In *LOCWEB '08: Proceedings of the first international workshop on Location and the web*, pages 49–56, New York, NY, USA. ACM.

[Girardin and Blat, 2007] Girardin, F. and Blat, J. (2007). Place this photo on a map: A study of explicit disclosure of location information. In *UbiComp 2007*.

[Goggles, 2008] Goggles, G. (2008). Website address. `http://www.google.com/mobile/goggles/`.

[Gokalp and Aksoy, 2007] Gokalp, D. and Aksoy, S. (2007). Scene classification using bag-of-regions representations. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8.

[Haralick et al., 1973] Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610 –621.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, 28:100–108.

[Hollenstein, 2008] Hollenstein, L. (2008). Capturing vernacular geography from georeferenced tags. *Masters Thesis, University of Zurich*.

[Howarth and Rger, 2004] Howarth, P. and Rger, S. (2004). Evaluation of texture features for content-based image retrieval. In Enser, P., Kompatsiaris, Y., OConnor, N. E., Smeaton, A. F., and Smeulders, A. W. M., editors, *Image and Video Retrieval*, volume 3115 of *Lecture Notes in Computer Science*, pages 2134–2135. Springer Berlin / Heidelberg.

[Hoxmeier et al., 2000] Hoxmeier, J. A., D, P., and Manager, C. D. (2000). System response time and user satisfaction: An experimental study of browser-based

applications. In *Proceedings of the Association of Information Systems Americas Conference*, pages 10–13.

[Hsu et al., 2002] Hsu, R.-L., Abdel-Mottaleb, M., and Jain, A. K. (2002). Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):696–706.

[Huang et al., 2005] Huang, W., Nakamori, Y., and Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.*, 32(10):2513–2522.

[Jain et al., 1997] Jain, A. K., Ratha, N. K., and Lakshmanan, S. (1997). Object detection using gabor filters. *Pattern Recognition*, 30:295–309.

[Jain and Vailaya, 1996] Jain, A. K. and Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244.

[Jones et al., 2010] Jones, Gareth J, F., Byrne, D., Hughes, M., Salway, A., and O'Connor, N. (2010). Automatic semantic annotation of landmark images with web mining. In *5th International Conference on Semantic and Digital Media Technologies*, SAMT 2010.

[Jschke et al., 2009] Jschke, R., Eisterlehner, F., Hotho, A., and Stumme, G. (2009). Testing and evaluating tag recommenders in a live system. In Benz, D. and Janssen, F., editors, *Workshop on Knowledge Discovery, Data Mining, and Machine Learning*, pages 44–51.

[Juan and Gwon, 2009] Juan, L. and Gwon, O. (2009). A comparison of sift, pca-sift and surf. *2009: International Journal of Image Processing*, 3(4):143–152.

[Kadir and Brady, 2001] Kadir, T. and Brady, M. (2001). Saliency, scale and image description. *Int. J. Comput. Vision*, 45:83–105.

[Kasutani and Yamada, 2001] Kasutani, E. and Yamada, A. (2001). The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. *Proceedings 2001 International Conference on Image Processing Cat No01CH37205*, 1(5):674–677.

[Kennedy et al., 2007] Kennedy, L., Naaman, M., Ahern, S., Nair, R., and Rattenbury, T. (2007). How flickr helps us make sense of the world: context and content in community-contributed media collections. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 631–640, New York, NY, USA. ACM.

[Kennedy and Naaman, 2008] Kennedy, L. S. and Naaman, M. (2008). Generating diverse and representative image search results for landmarks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 297–306.

[Kim et al., 2005] Kim, W., Yoo, S. J., Kim, J.-s., Nam, T. Y., and Yoon, K. (2005). Detecting adult images using seven mpeg-7 visual descriptors. In Shimojo, S., Ichii, S., Ling, T. W., and Song, K.-H., editors, *Web and Communication Technologies and Internet-Related Social Issues - HSI 2005*, volume 3597 of *Lecture Notes in Computer Science*, pages 336–339. Springer Berlin / Heidelberg.

[Konolige et al., 2009] Konolige, K., Bowman, J., Chen, J. D., Mihelich, P., Calonder, M., Lepetit, V., and Fua, P. (2009). View-based maps. In *Proceedings of Robotics: Science and Systems*, Seattle, USA.

[Kotoulas and Andreadis, 2003] Kotoulas, L. and Andreadis, I. (2003). Colour histogram content-based image retrieval and hardware implementation. *Circuits, Devices and Systems, IEE Proceedings -*, 150(5):387–93.

[Kuthan and Hanbury, 2006] Kuthan, S. and Hanbury, A. (2006). Hierarchical image classification. *imageval.org 2006*.

[Lamrous and Taileb, 2006] Lamrous, S. and Taileb, M. (2006). Divisive hierarchical k-means. In *Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, pages 18–, Washington, DC, USA. IEEE Computer Society.

[Lebanon et al., 2007] Lebanon, G., Mao, Y., and Dillon, J. (2007). The locally weighted bag of words framework for document representation. *J. Mach. Learn. Res.*, 8:2405–2441.

[Lee et al., 2010] Lee, D. C., Ke, Q., and Isard, M. (2010). Partition min-hash for partial duplicate image discovery. In *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, pages 648–662, Berlin, Heidelberg. Springer-Verlag.

[Li et al., 2009a] Li, X., Snoek, C. G. M., and Worring, M. (2009a). Annotating images by harnessing worldwide user-tagged photos. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '09, pages 3717–3720, Washington, DC, USA. IEEE Computer Society.

[Li et al., 2009b] Li, Y., Crandall, D., and Huttenlocher, D. (2009b). Landmark classification in large-scale image collections. In *ICCV09*, pages 1957–1964.

[Lin and Nevatia, 1998] Lin, C. and Nevatia, R. (1998). Building detection and description from a single intensity image. *Comput. Vis. Image Underst.*, 72(2):101–121.

[Lindeberg, 1994] Lindeberg, T. (1994). Scale-space theory in computer vision.

[Liu et al., 2009] Liu, T., Liu, J., Liu, Q., and Lu, H. (2009). Expanded bag of words representation for object classification. In *Proceedings of the 16th IEEE*

international conference on Image processing, ICIP'09, pages 297–300, Piscataway, NJ, USA. IEEE Press.

[Lowe, 1999] Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision - Volume 2*, pages 1150–1157.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

[Ma et al., 2009] Ma, X., Wu, J.-S., Liu, H.-D., Yang, X.-N., Xie, J.-M., and Sun, X. (2009). Svm-based approach for predicting dna-binding residues in proteins from amino acid sequences. In *IJCBS '09: Proceedings of the 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*, pages 225–229, Washington, DC, USA. IEEE Computer Society.

[MacQueen, 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

[Mahapatra et al., 2011] Mahapatra, A., Wan, X., Tian, Y., and Srivastava, J. (2011). Augmenting image processing with social tag mining for landmark recognition. In *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part I*, MMM'11, pages 273–283, Berlin, Heidelberg. Springer-Verlag.

[Manjunath and Ma, 1996] Manjunath, B. S. and Ma, W. Y. (1996). Texture Features for Browsing and Retrieval of Image Data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842.

[Manjunath et al., 2001] Manjunath, B. S., Ohm, J. R., Vinod, V. V., , and Yamada, A. (2001). Color and texture descriptors. *IEEE Trans. Circuits and Systems for Video Technology, Special Issue on MPEG-7*, 11(6):703–715.

[Manjunath et al., 2002] Manjunath, B. S., Salembier, P., and Sikora, T., editors (2002). *Introduction to MPEG-7: Multimedia Content Description Language*. Wiley.

[Messing et al., 2001] Messing, D. S., Beek, P., and Errico, J. H. (2001). The mpeg-7 colour structure descriptor: Image description using color and local spatial information. In *In: Proc. Internat. Conf. on Image Processing*, pages 670–673.

[Metzler, 2008] Metzler, D. (2008). Beyond bags of words: effectively modeling dependence and features in information retrieval. *SIGIR Forum*, 42(1):77–77.

[Missaoui et al., 2004] Missaoui, R., Sarifuddin, M., and Vaillancourt, J. (2004). An effective approach towards content-based image retrieval. In *CIVR*, pages 335–343.

[Moëllic et al., 2008] Moëllic, P. A., Haugeard, J. E., and Pitel, G. (2008). Image clustering based on a shared nearest neighbors approach for tagged collections. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, New York, NY, USA. ACM.

[MovableType, 2001] MovableType (2001). Website address. `http://www.movabletype.org`.

[Murillo et al., 2007] Murillo, A., Guerrero, J., and Sagues, C. (2007). Surf features for efficient robot localization with omnidirectional images. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3901 –3907.

[MySpace, 2003] MySpace (2003). Website address. `http://www.myspace.com`.

[Nister and Stewenius, 2006] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2161–2168.

[O'Hare et al., 2005] O'Hare, N., Gurrin, C., Jones, G., and Smeaton., A. F. (2005). Combination of content analysis and context features for digital photograph retrieval. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, pp323-328*, pages 323–328.

[Ojala et al., 2001] Ojala, T., Rautiainen, M., Matinmikko, E., and Aittola, M. (2001). Semantic image retrieval with hsv correlograms. In *Proc. 12th Scandinavian Conference on Image Analysis*, pages 621–627.

[O'Reilly, 2005] O'Reilly, T. (2005). What is web 2.0? retrieved on 20th july 2010. `http://oreilly.com/web2/archive/what-is-web-20.html`.

[Panoramio, 2005] Panoramio (2005). Website address. `http://www.panoramio.com`.

[Pavlenko, 2003] Pavlenko, T. (2003). On feature selection, curse-of-dimensionality and error probability in discriminant analysis. *Journal of Statistical Planning and Inference*, 115(2):565–584.

[Peterson and Larin, 2009] Peterson, L. E. and Larin, K. V. (2009). Image classification of artificial fingerprints using gabor wavelet filters, self organising maps and hermite laguerre neural networks. *Int. J. Knowl. Eng. Soft Data Paradigm.*, 1:239–256.

[Petkovic et al., 1996] Petkovic, D., Niblack, W., Flickner, M., Steele, D., Lee, D., Yin, J., Hafner, J., Tung, F., Treat, H., Dow, R., Gee, M., Vo, M., Vo, P., Holt, B., Hethorn, J., Weiss, K., Elliott, P., and Bird, C. (1996). Recent applications of

ibm's query by image content (qbic). In *SAC '96: Proceedings of the 1996 ACM symposium on Applied Computing*, pages 2–6, New York, NY, USA. ACM.

[Philbin et al., 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8.

[Philbin et al., 2008] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Ping-Feng and Chih-Sheng, 2005] Ping-Feng, P. and Chih-Sheng, L. (2005). A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505.

[Popescu and Mollic, 2009] Popescu, A. and Mollic, P.-A. (2009). Monuanno: automatic annotation of georeferenced landmarks images. In Marchand-Maillet, S. and Kompatsiaris, Y., editors, *CIVR*. ACM.

[Qamra and Chang, 2008] Qamra, A. and Chang, E. Y. (2008). Scalable landmark recognition using extent. *Multimedia Tools Appl.*, 38(2):187–208.

[Qingji et al., 2008] Qingji, G., Juan, L., and Guoqing, Y. (2008). Vision based road crossing scene recognition for robot localization. *Computer Science and Software Engineering, International Conference on*, 6:62–66.

[Rafiee and Sarajian, 2008] Rafiee, A. and Sarajian, M. R. (2008). Classification of buildings and roads using support vector machine. In *DICTA '08: Proceedings of the 2008 Digital Image Computing: Techniques and Applications*, pages 111–116, Washington, DC, USA. IEEE Computer Society.

[Rahmani et al., 2008] Rahmani, R., Goldman, S. A., Zhang, H., Cholleti, S. R., and Fritts, J. E. (2008). Localized content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1902–1912.

[Reddit, 2005] Reddit (2005). Website address. `http://www.reddit.com`.

[Salway et al., 2010] Salway, A., Kelly, L., Skadiņa, I., and Jones, G. J. F. (2010). Portable extraction of partially structured facts from the web. In *Proceedings of the 7th international conference on Advances in natural language processing*, IceTAL'10, pages 345–356, Berlin, Heidelberg. Springer-Verlag.

[Sanderson and Kohler, 2005] Sanderson, M. and Kohler, J. (2005). Analyzing geographic queries. In *Proc. of the Workshop on Geographic Information Retrieval*.

[Schindler et al., 2007] Schindler, G., Brown, M., and Szeliski, R. (2007). City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7.

[Sigurbornsson and van Zwol, 2008] Sigurbornsson, B. and van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336. ACM.

[Singh et al., 2003] Singh, S. K., Chauhan, D. S., Vatsa, M., and Singh, R. (2003). A robust skin color based face detection algorithm, tamkang. *Journal of Science and Engineering*, 6:227–234.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477 vol.2.

[Sivic and Zisserman, 2006] Sivic, J. and Zisserman, A. (2006). Video Google: Efficient visual search of videos. In Ponce, J., Hebert, M., Schmid, C., and

Zisserman, A., editors, *Toward Category-Level Object Recognition*, volume 4170 of *LNCS*, pages 127–144. Springer.

[Smeulders et al., 2000] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380.

[Smith and fu Chang, 1996] Smith, J. R. and fu Chang, S. (1996). Visualseek: a fully automated content-based image query system. In *MULTIMEDIA '96 Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98.

[Snavely et al., 2006] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, New York, NY, USA. ACM.

[Sparck Jones, 1988] Sparck Jones, K. (1988). A statistical interpretation of term specificity and its application in retrieval. In *Document retrieval systems*, pages 132–142, London, UK, UK. Taylor Graham Publishing.

[Stricker and Orengo., 1995] Stricker, M. A. and Orengo., M. (1995). Similarity of color images. In *Proc. Storage and Retrieval for Image and Video Databases III*.

[Szummer and Picard, 1998] Szummer, M. and Picard, R. W. (1998). Indoor-outdoor image classification. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, pages 42–51.

[Takeuchi and Hebert, 1998] Takeuchi, Y. and Hebert, M. (1998). Evaluation of image-based landmark recognition techniques. Technical Report CMU-RI-TR-98-20, Robotics Institute, Pittsburgh, PA.

[Talbar and Varma, 2010] Talbar, S. N. and Varma, S. L. (2010). Article:color spaces for transform-based image retrieval. *International Journal of Computer Applications*, 9(12):4–6. Published By Foundation of Computer Science.

[Tamura et al., 1973] Tamura, H., Mori, S., and Yamawaki, T. (1973). Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460 –473.

[TechCrunch, 2010] TechCrunch (2010). Flickr gets more photogenic with a complete photo page overhaul. retrieved on 3rd august 2010. `http://techcrunch.com/2010/06/23/new-flickr-design/`.

[Tirilly et al., 2008] Tirilly, P., Claveau, V., and Gros, P. (2008). Language modeling for bag-of-visual words image categorization. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 249–258, New York, NY, USA. ACM.

[Torkkola, 2002] Torkkola, K. (2002). Discriminative features for document classification. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 472–475 vol.1.

[Torralba et al., 2003] Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). Context-based vision system for place and object recognition. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 273, Washington, DC, USA. IEEE Computer Society.

[Toyama et al., 2003] Toyama, K., Logan, R., and Roseway, A. (2003). Geographic location tags on digital images. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 156–166, New York, NY, USA. ACM.

[Tsapatsoulis and Theodosiou, 2009] Tsapatsoulis, N. and Theodosiou, Z. (2009). Object classification using the mpeg-7 visual descriptors: An experimental evaluation using state of the art data classifiers. In Alippi, C., Polycarpou, M., Panayiotou, C., and Ellinas, G., editors, *Artificial Neural Networks ICANN*

2009, volume 5769 of *Lecture Notes in Computer Science*, pages 905–912. Springer Berlin / Heidelberg.

[Tuffield et al., 2006] Tuffield, M. M., Harris, S., Brewster, C., Gibbins, N., Ciravegna, F., Sleeman, D., Shadbolt, N. R., and Wilks, Y. (2006). Image annotation with photocopain. In *In Proceedings of Semantic Web Annotation of Multimedia (SWAMM-06) Workshop at the World Wide Web Conference 06. WWW*, pages 22–26.

[Vailaya et al., 1998] Vailaya, A., Jain, A., and Zhang, H. J. (1998). On image classification: City images vs. landscapes. *PATTERN RECOGNITION*, 31:1921–1935.

[van de Sande et al., 2010] van de Sande, K. E., Gevers, T., and Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1582–1596.

[van Gemert et al., 2010] van Gemert, J. C., Snoek, C. G. M., Veenman, C. J., Smeulders, A. W. M., and Geusebroek, J. M. (2010). Comparing compact codebooks for visual categorization. *Computer Vision and Image Understanding*, 114(4):450–462.

[van Rijsbergen, 1979] van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, 2 edition.

[Viitaniemi and Laaksonen, 2009] Viitaniemi, V. and Laaksonen, J. (2009). Combining local feature histograms of different granularities. In *Proceedings of the 16th Scandinavian Conference on Image Analysis*, SCIA '09, pages 636–645, Berlin, Heidelberg. Springer-Verlag.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511.

[W3C, 2006] W3C (2006). WGS84 geo positioning. http://www.w3.org/2003/01/geo/wgs84_pos.

[Walter et al., 2008] Walter, B., Bala, K., Kulkarni, M., and Pingali, K. (2008). Fast agglomerative clustering for rendering. In *IEEE Symposium on Interactive Ray Tracing (RT)*, pages 81–86.

[Weldon and Higgins, 1999] Weldon, T. P. and Higgins, W. E. (1999). Designing multiple gabor filters for multitexture image segmentation. *Optical Engineering*, pages 1478–1489.

[Wikipedia, 2001] Wikipedia (2001). Website address. `http://www.wikipedia.org`.

[WordNet, 1985] WordNet (1985). Website address. `http://wordnet.princeton.edu`.

[WordPress, 2003] WordPress (2003). Website address. `http://www.wordpress.org`.

[Xiao et al., 2010] Xiao, X., Xu, C., and Wang, J. (2010). Landmark image classification using 3d point clouds. In *Proceedings of the international conference on Multimedia*, MM '10, pages 719–722.

[Xu and Zhang, 2005] Xu, Y. and Zhang, X. (2005). Gabor filterbank and its application in the fingerprint texture analysis. *Parallel and Distributed Computing Applications and Technologies, International Conference on*, 0:829–831.

[Yahoo, 1995] Yahoo (1995). Website address. `http://www.yahoo.com`.

[Yan et al., 2003] Yan, R., Liu, Y., Jin, R., and Hauptmann, A. (2003). On predicting rare classes with svm ensembles in scene classification. In *In ICASSP*, pages 21–24.

[Yan-Tao Zheng, 2009] Yan-Tao Zheng, Ming Zhao, Y. S. H. A. U. B. A. B. F. B. S. C. H. N. (2009). Tour the world: building a web scale landmark recognition engine. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*.

[Yeh et al., 2004] Yeh, T., Tollmar, K., and Darrell, T. (2004). Searching the web with mobile images for location recognition. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, CVPR'04, pages 76–81, Washington, DC, USA. IEEE Computer Society.

[YouTube, 2005] YouTube (2005). Website address. `http://www.youtube.com`.

[Zhang and Kosecka, 2007] Zhang, W. and Kosecka, J. (2007). Hierarchical building recognition. *Image and Vision Computing*, (5):704–716.