ATIF USMAN
01-134122-021

Estimation of geographic information from a single image

Bachelor of Science in Computer Science

Supervisor: Dr. Muhammad Muzammil

Department of Computer Science
Bahria University, Islamabad

May 2016

Certificate

We accept the work contained in the report titled "Estimation of geographic information from a single image", written by Mr. Atif Usman as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Dr. Muhammad Muzammil (Assistant Professor)

_____

Internal Examiner: Mr. Humayun Zaheer (Assistant Professor)

_____

External Examiner:

_____

Project Coordinator: Dr. Arif-Ur-Rehman (Assistant Professor)

_____

Head of the Department: Dr. Faisal Bashir (Associate Professor)

_____

May 19^{th}, 2016

Abstract

With the internet rapidly turning into a hub of available information resource, social networking sites have seen an unprecedented increase in their popularity during the past decade leading to millions of users joining from around the world. Some of these social networking sites are dedicated to only image and video content based experience sharing. Flickr.com is one of these popular image sharing sites owned by yahoo group where each day millions of images are shared online from people around the world. Some of these images are even available with a very precise information about their location. Considering this, it is becoming more and more feasible for us to start utilizing this information in form of images for enhanced future of the computer vision field. These available images with known locations can be used to determine or predict the location of images whose location is not already known. This report covers the details about a proposed system which can be used to estimate the geographic information of images with unknown origin based on the available data of images with precisely known locations. Different high and low level image processing techniques will be utilized in this regard which are discussed in detail in the coming chapters of this report. The idea is to extract useful features from images in training data with known location; and compare them with features of test data images whose location is to be estimated. By doing so, we will be able to find an approximate match of test data image in training data images, based on this match; some estimations can be made about the location of test data images.

Acknowledgments

The world suddenly becomes a better place to live in when we start to appreciate what we have.

Dedicated to my family, teachers, friends and all the people that I've had something to learn from.

To begin with, I am very grateful to my supervisor Dr. Muhammad Muzammil for his guidance throughout the whole development process of this project. He has always been very helpful and cooperative and the completion of this project would not have been such a success without all the help and guidance that he provided. I am also highly thankful to all the teachers I've ever had who helped me become a better person over the years.

I am equally indebted to Bahria University Islamabad campus for supporting me throughout the project.

I dedicate this thesis to my family. Everything that I have achieved is because of my parent's prayers; the greatest gift in this world. The most prized jewel one possesses in this world is one's family. To my father, who inspires me; to my mother whose love and affection is unmatched and to my siblings whom I love very dearly.

And at last but not the least I thank Almighty Allah for all the blessings and strength that He bestowed upon me to carry out this task, for without Allah's will nothing is possible. May Allah grant us strength and will to fulfill our dreams and to play our part in making this world a better place for the good of humanity and also to appreciate and take care of this beautiful planet that we have been blessed with.

ATIF USMAN
Islamabad, Pakistan

September 2015

"We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn't anyone else."

Regina Dugan

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

WCF	Windows Communication Foundation
CBIR	Content-based image retrieval
DIP	Digital Image Processing
JSON	JavaScript Object Notation
API	Application program interface
DB	Database
UNICODE	Unique, Universal, and Uniform Character enCoding
XML	Extensible Markup Language
HTML	HyperText Markup Language

Chapter 1

Introduction

1.1 Overview

To give an overview, it can be stated that this project is focused on using visual information from a single image for an attempt to estimate the geographic information of the location where that given image was captured using comparison approach. This project and those similar to it have a variety of application contexts, it can not only be utilized for the most general use as to finding out where an old forgotten picture might have been taken but these type of projects are also necessary for the advancements in the field of robotics and computer vision. The idea of comparing visual information to estimate location is coming closer to reality due to the advent of global socialization leading to surge of visual data available over the internet resulting in the large repository of images open to public whose geographic information is already known.

1.2 Project Description

The idea originally published by James Hayes [1] is inspired from one of the most remarkable and unique instinct of perception that humans possess which involves semantic reasoning of what we see around us. When looking at a photograph humans tend to make some basic assumptions about the context of the photograph as to what could be the possible whereabouts of that photograph. For instance if it is a landscape photograph we try to analyze it by noticing some basic features available like the terrain, surroundings or the weather, even if it's from somewhere completely unknown or someplace we've never been to or seen before. We are able to efficiently separate pictures of rainforests from deserts, pictures of grasslands from plateaus, pictures of seashores from mountain ranges and it all happens instantly. The ability to do this involves having knowledge from

past experiences. The reason we are able to do this is because we have the ability to learn from past experiences and by comparing the information available in the photograph with the knowledge we already have, we are able to judge the context of that photograph to extraordinary levels of accuracy. We have this knowledge from past that we are constantly learning throughout the entirety of our lifespans. This project is a small step further in the field of computer vision and aims to incorporate this remarkable human feature of perceiving environment quiet accurately to future computers or robots not only that, it also serves as a helpful tool as well for general users around the world for remembering an old forgotten photograph or for security purposes as to finding locations in the photographs.

However, the concept of completely incorporating this powerful human capability to computers seems pretty farfetched and unlikely. The machines of today as powerful as they may be are still far more inferior compared to the human intelligence which is a key component for perception and semantic reasoning. But what is possible for computers today is to adopt a pure data driven approach to compare a given photograph with and already available inventory of photographs and find the closest match which naturally; in an ideal situation, would be another photograph of the same place where the photograph under question was taken. In order to for a machine to be able to do that, a considerably large amount of visual data must be available in the inventory. Which ideally should cover the whole planet but that is not possible at least for now.

Nevertheless, with the arrival of social networking each day millions of images are shared over the internet from people around the world. Some of these images are even available with a very precise information about their location. This huge repository of available images could be a step forward in machine learning and development of systems which could prove to be very useful for future robotics in an effort to make them perceptive of their surroundings.

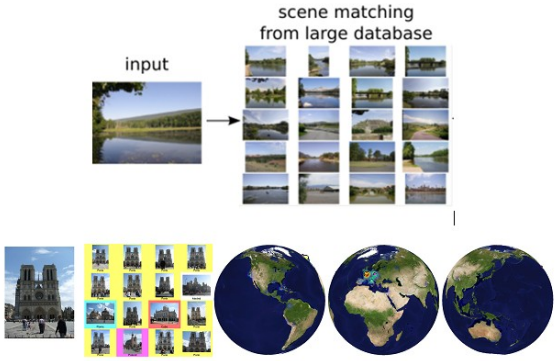

Figure 1.1: Illustration of scene matching to estimate location

Using some efficient scene matching algorithm the information about an unknown image can be derived in terms of its nearest neighbors or matches found in the available

known image data, if a good match is found it can be safe to assume that the image came from same locality as the one it matched with.

1.3 Project Objectives

In order to find possible matches for a given unknown image, the project follows a data intensive approach as it requires processing large dataset of already known images from the internet for brute-force scene matching. For the dataset of required images 'Flickr.com' is the most suitable candidate. Owned by Yahoo group Flickr is a social networking site which mainly focuses on visual posts mostly in form of photographs. Users from around the world post photographs on Flickr for sharing their experience, a considerable portion of these photograph also come with precise information about their location. The whole project can be divided into following steps which will be further discussed in detail in the later chapters of this report.

- The project first involves getting data from Flickr.com using its API, for that purpose a we have designed and developed a small desktop application which takes essential inputs from user including
  - The user's Flickr API key which entitles user to the use of data available on Flickr domain.
  - The tags associated with the data to be downloaded. These are used to specify the place or location to which the query must be focused to get the relevant pictures posted by Flickr users with those tags. Tags can include name of a famous tourist place or a city or even a whole country. The data will be returned against the tags entered by the user.
  - The query output path as to where exactly must be data downloaded on target user machine.
- For the ease of application and demonstration and due to limited resources the dataset involves 4 different famous locations and hundreds of images will be used from these locations in order to yield better results.
- After acquiring dataset using Flickr downloader the GIST descriptor was computed for all the images in the dataset and uploaded to the cloud dataset for distributed availability and performance.
- Once an unknown input is provided, to compare the features of this unknown image with the known dataset; a GIST Descriptor feature vector for the target image is computed and compared with the feature vectors of each of the known images.

- Consequently, after comparison the classifier assigns the target image to the closest match in the data set. In this way making significant assumptions about the geographic information of the target image will be possible based on the information available for its nearest neighbor in the known dataset.

1.4 Project Scope

The scope of the project will largely depend on the evidence in the image database available. The project implementation will be as good as the images available. The idea that every image taken can be traced back to some location using the available image repository is still farfetched. For this to be possible, huge amount of visual data with information will be required and we are still a long way from that to happen. The huge repository of images floating over the internet is still very far from enough to cover every city, every street, every forest land, every desert, every river, every lake or even every country or island on the planet. Although the features and techniques involved in the scene matching approach also matter a lot but this project is research based and requires a lot of resources to be invested to get the desired results. With the passage of time as the technology evolves considering the rate at which it is advancing we are going to see remarkable improvements in the computer vision field and in the advanced robotics.

Chapter 2

Literature Review

This chapter covers a little detailed overview of some of the existing systems which are currently using scene matching approaches to categorize images according to their similarities with other images, as well as the libraries and algorithms that are useful for the image processing and image analysis. Some popular search engines including google; allow users to search their queries using images as input rather than text. These search engines then use different scene matching approaches to search for the results for the query similar to the input image.

2.1 Background

Localization of images on a geographical map depending on their context has been one of the most important and difficult problems in computer vision. But as already mentioned in previous chapter the solution becomes exponentially less difficult with increased evidence in the data sources that are available. Jacobs et al [2] proposed a simple method which involved geolocating a webcam based on its video stream correlated with weather maps from satellites in the course of same time period.

In terms of an approach for the solution of this problem the work is quiet similar to what James Hayes [1] suggests in his paper regarding the algorithm and data repository source i.e. Flickr.com. Due to this availability of GPS-tagged images from all over the world combined with advancements in effective and efficient image feature calculations a number of scientists and developers have tried developing place recognition algorithms.

The most important question here is whether the available data will ever be enough to cover the whole planet, which can't still be answered with certainty. The idea is far more complex than it seems, the famous landmarks might have uncountable pictures available online but they cover less than even the fraction of the entire planet. In reality covering the

entire planet is very unlikely, not just because of computational cost, but simply because it is also impossible in terms of manpower, energy and cost. Common people are not interested in photographing and posting every scene in their surroundings all the time. In the future we may be able to design energy and cost efficient dedicated systems that could be used to serve this special purpose of collecting visual data from around the globe similar to what google is trying to do currently by deploying rovers and high definition equipment all over the world to cover different places for their google street view program.

This also brings us to a practical example of this project; google image search. Google Images has a deployed Search by Image feature for users to perform reverse image searches meaning that users are not only able to search for images through text inputs, but they are also able to search textual or visual information by giving an image as an input. Unlike the traditional search mechanism which involve textual query for results, These type of systems utilize what is more commonly known as the content-based image retrieval (CBIR) query technique which involves providing the CBIR system with an unknown sample image it will then base its search upon; in terms of information retrieval, the sample image is what formulates a search query. The CBIR system deploys several image processing and feature extraction techniques to compare images and retrieve most relevant results based on the similarity with the input image [3].

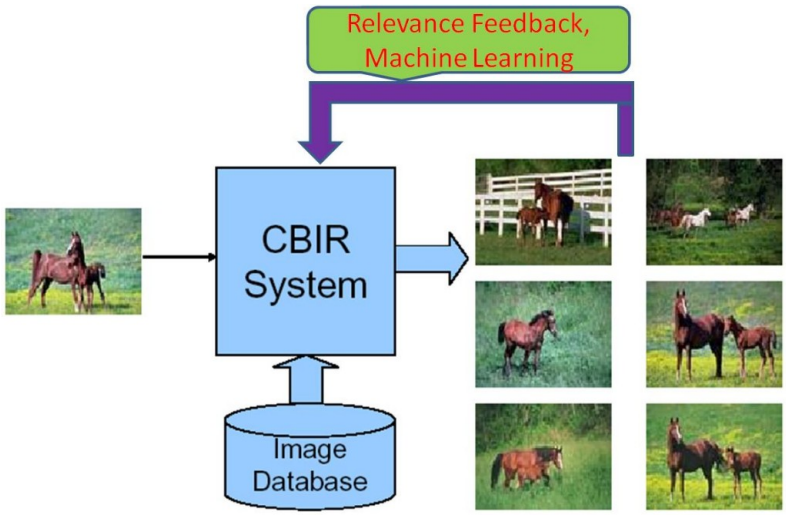

Figure 2.1: Image search based on query image content

It is one of the most effective systems of its kind available yet and google uses the most advanced algorithms to develop a mathematical model of the input image and comparing it with billions of images in its own database gathered from all over the internet and not just from social networking sites. Google accomplishes this by analyzing the submitted picture and adopting following basic steps for result generation.

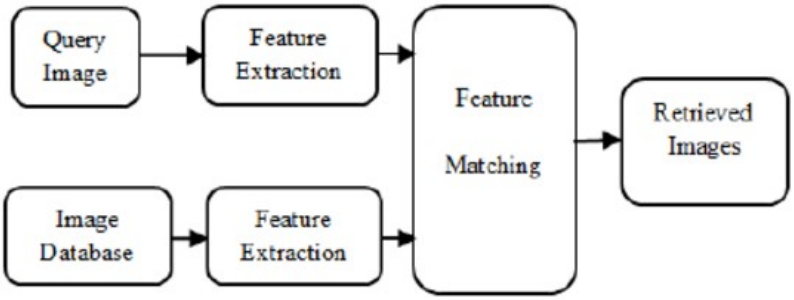

Figure 2.2: Block diagram showing CBIR system sequence

- Analyze image - The input image is processed to compute feature identifiers like points, textures, lines, and colors.
- Generate query – The unique features of the given image/photograph are used to produce a search query.
- Match image - The query is compared to billions of images in Google's back end database to find the most similar match.
- Return results - Google's matching algorithm return matching image along with other images that are visually similar as results to users.

Another famous CBIR platform available over the internet for the users around the globe is called TinEye.com. TinEye.com is another search engine that provides the functionality of getting query results through an image input rather than textual query. It is the first search engine of its kind that provides strictly image based query submission capabilities with no keywords involved.

Although the exact algorithm working behind the TinEye.com for matching images has never been disclosed publicly, many known techniques or algorithms are known to be part of TinEye image search one of these famous algorithm was proposed by Dr. Neal Krawetz[4]

The accuracy of TinEye.com allows it to search even most edited pictures online which is why it is quiet popular among users. As of 2016, TinEye.com claims to have 15.1 billion images indexed in its database which is constantly growing.

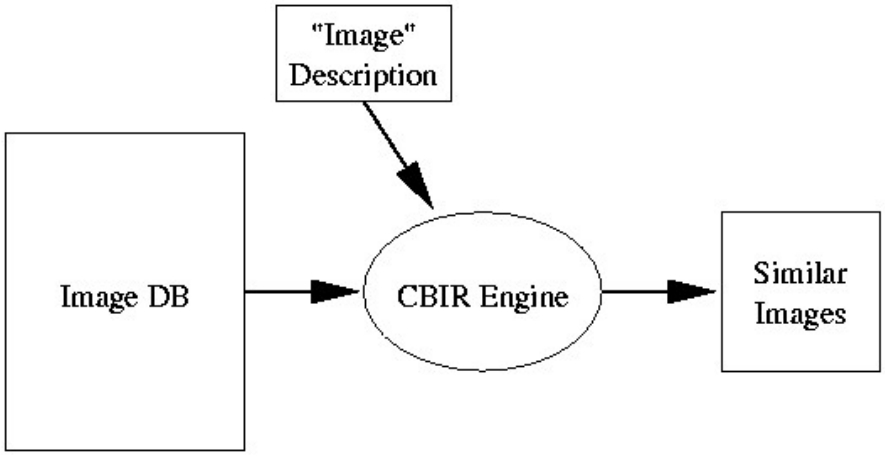

Figure 2.3: Simple representation of CBIR engine

Chapter 3

Requirement Specifications

All the systems mentioned in the previous chapter tend to have a slightly limited context despite having unparalleled potential. They work on a much broader level which is to implement CBIR system and find matches of an input image regardless of its context or details. The system proposed in this project tends to focus on a particular category of those query images i.e. outdoor photographs of places and landscapes for the sole purpose of geographically mapping those images as per their location.

This chapter is intended to cover a detailed account of system requirements necessary for the project implementation. As established earlier in order to reason about the global location of an arbitrary scene we first need a large number of suitable images that are labelled with geographic information for comparison. This information could be in the form of text keywords or it could be in the form of GPS coordinates. As already mentioned in the introduction portion, the system will work when a user will provide an input image of a place with unknown location or origin and the system will estimate the geographic information of the location based on the input image by matching and finding the nearest neighbor in the database using scene matching approach. For this purpose a large database of geotagged images is required to obtain more reliable results. Considering the technological restrains for this project data of some selected places will be used to demonstrate the effectiveness of the application. The data will be divided into several bins depending on the number of locations and the project will take a previously unknown image belonging to one of the selected places the system will then try to determine the bin that image must belong to.

3.1 Image Database

For the development of database fortunately there is a huge (and rapidly growing) amount of online images with geotagged information of their location. Image sharing is becoming more popular each day and the amount of visual data online is rapidly increasing. Some of the social image sharing websites allow users to geotag their pictures and upload them publicly for the purpose of sharing their experiences.

Flickr.com which is owned by the yahoo group is one of these famous social networking sites. Flickr is an image and video hosting website, and web services suite that was created by Ludicorp in 2004 and acquired by Yahoo in 2005. In addition to being a popular website for users to share their personal and public photographs as an online community, the service is also widely used by developers and experts for research purposes in the field of image processing.

Flickr.com as of now is home to millions of pictures with geographic information stored with them by the users. But in order to obtain a useful, high-quality database based on user collected and labelled content some form of filtering is still required for non-essential images which might disrupt the whole outcome of the system. For instance, people taking and uploading pictures of themselves near pyramids and geotagging them with the location of pyramids results in non-essential or even wrong result generating data for the system.

Due to these reasons in most cases pictures taken only by tourists are most suitable because they often try to capture unique and interesting attributes of a place. Many of these images can be found because they often have geographic keywords associated with them (i.e. city or country names). In most cases the images are also of poor quality (low resolution, noisy, filtered, edited with effects) or depict scenes which are not entirely useful for geolocation such as self-portraits, abstracts, and macro photography.

3.2 Software and additional Requirements

As the database for this kind of project will require thousands of pictures of a single location for better results, increasing the number of locations exponentially increases the size of required database and processing it requires, not to mention it constant availability. For this purpose the database used in this project is cloud based and the whole database is stored in the Microsoft Azure cloud database in SQL. This allows for it to be accessed from anywhere in the world at any given time. Also this could help in keeping the consistent flow of data for application user anywhere in the world. Simply updating database will allow users to access this data from anywhere.

In addition to all this there are some concrete software requirements as well for the proper running of the application. As the feature calculation is Matlab based and is integrated with the C# front end the program requires a running instance of Matlab

Runtime Compiler. Also the product is developed on .Net framework 4.5 and is necessary for running the application.

The system also requires a live internet connection for two purposes:

- To access database on cloud of already stored image data for comparison with unknown image.
- To display the google map on application front end where the image location is going to be pinned.

Apart from all these necessary requirements there are some technological or hardware requirements as well the processing power of the machine and memory available do play an important role. The better these attributes of the running machine the better will be the performance of the application.

In terms of hardware requirements one feature of the application also requires an android device with camera to capture images on runtime for finding location. This feature allows user to connect their android devices with the desktop application using wireless connection through Wifi and transfer live stream of camera to the desktop application and capture images on the runtime for evaluation.

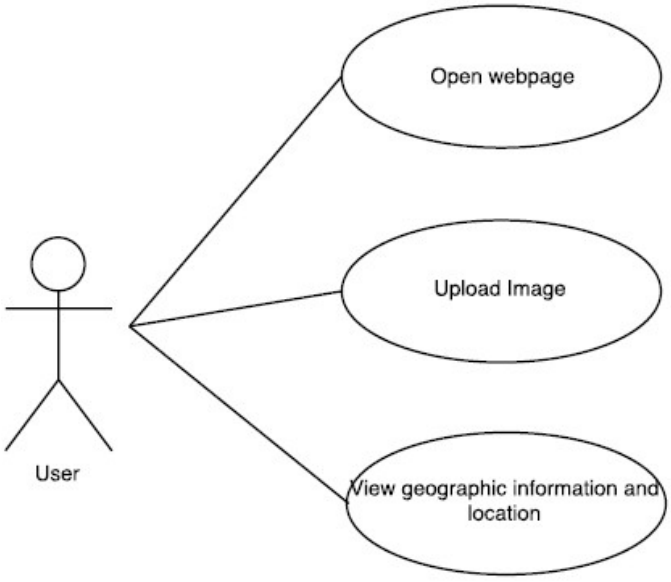

Figure 3.1: Use Case diagram showing functional requirements of the system

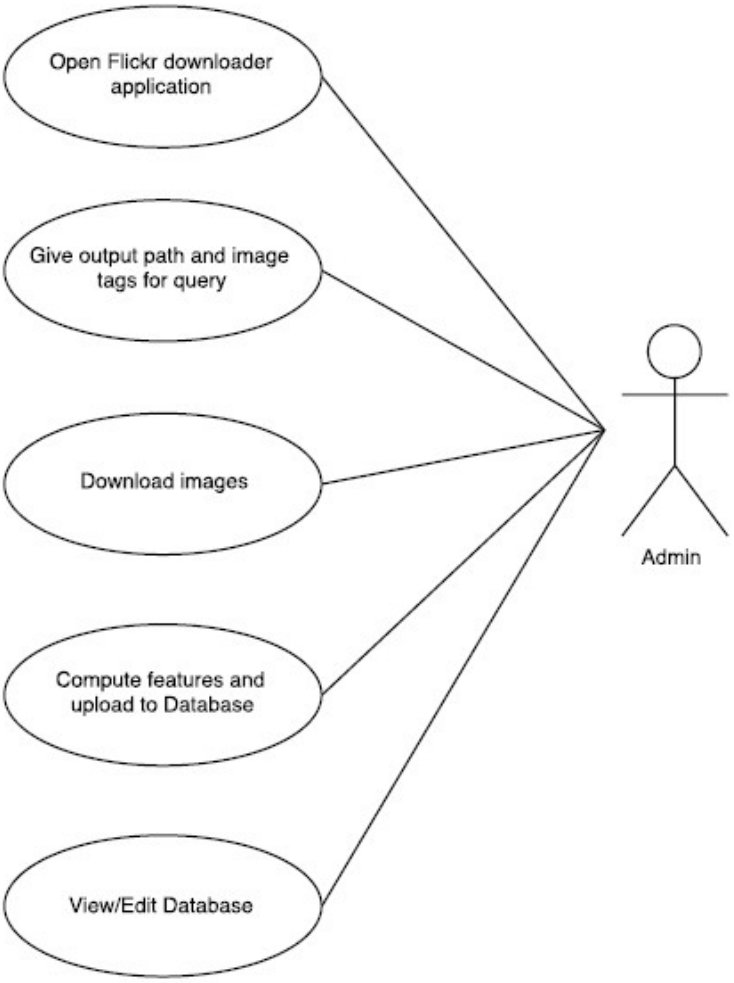

Figure 3.2: Use Cases for Admin Role

Chapter 4

Design

This chapter covers the design phase of the whole system. The requirements as already mentioned in previous chapter must be brought together and consumed efficiently in order to design and implement a working system with lowest possible levels of shortcomings or errors.

4.1 System Architecture

Considering a high level of architecture abstraction for this project the project can be divided in to two separate sub systems one consisting of data downloading and database composing sub system and second sub system with the whole location determining system with the user front end. The WCF service part ensures a single common point for feature extraction and helps in keeping the system consistent with different front end platforms such as web and desktop user front ends.

Following diagrams show the basic architecture of both subsystems. The working and structural details of these modules have been discussed thoroughly in the next chapter, so this chapter only covers the high level view of the whole system from different contexts.

This figure is the high level architecture of the desktop application system used to retrieve images from Flickr.com and then storing the database for the main project module to utilize.

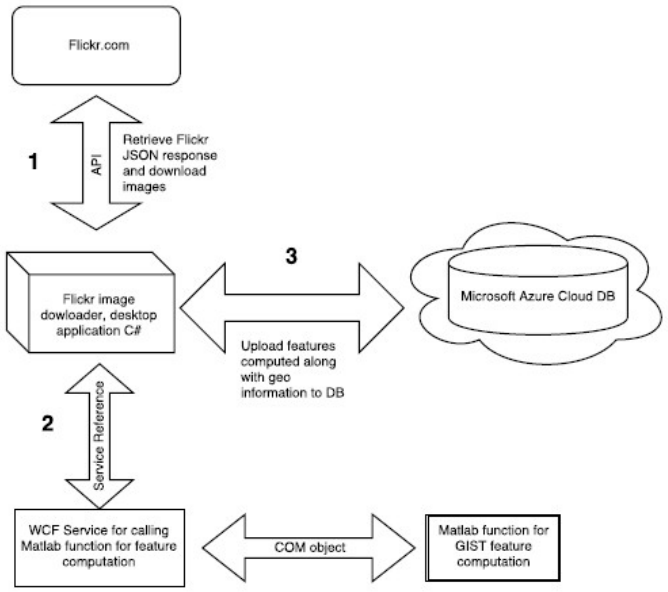

Figure 4.1: Architecture of Flickr module for downloading image data from Flickr API.

The next figure shows the architecture of the main module of the project which takes in user input in form of query image and compares it with database entries to determine its geographic information. The details of both architectures along with their implementation and working are discussed in the next chapter.

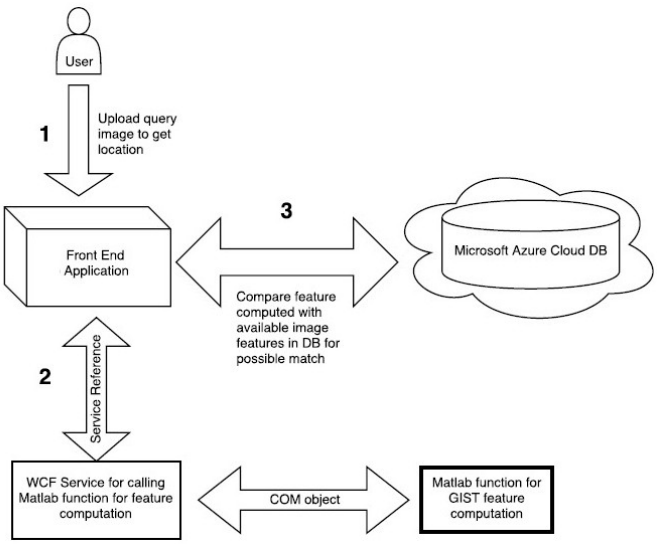

Figure 4.2: Architecture of the main module of the system for geolocating images.

4.2 Design Constraints

With the basic architecture of the whole system being discussed, the constraints and limitations of the system must also be addressed in order to better understand the implementation of the system. One of the most dominating constraints in a project of this kind is undoubtedly the amount of available evidence in the data. As already mentioned on numerous points in the previous chapters, the project implements a pure data driven approach with results depending on the available data, so the efficiency of the system most depends on the data inventory. But of course with increase in data, another technical factor comes to light as the computational cost also hikes up which requires highly powerful system for normal performance.

The factors mentioned above must be addressed when building the system on a much larger scale or for commercial use. However for this project we limit the scope to selected locations and implement system on a local computer with limited resources. Considering all that, the biggest constraint for this system is the limited integration level between Matlab and other development platform like the one used in this project asp.NET and C# language. Matlab is an independent platform for complex engineering and research problems it is almost impossible to entirely integrate it with other platforms. The approach used in this project is the use of COM object [5][6] and Microsoft's matlab library which allows generating a running Matlab instance and call specific functions without the ability to pass complex object like images which was needed in this project and hence different methods were adapted the details of which are discussed in the chapter 5 of this report. Nevertheless Matlab is a rich environment which provides a vast variety of functionalities to be performed for research purposes and is ideal for the field of image processing which is why feature computation was done through Matlab.

4.3 Design Methodology

Summarize the approach that will be used to create and evolve the designs for this system. Cover any processes, conventions, policies, techniques or other issues which will guide design work. This is for deciding whether you will use structured, object-oriented or other specific methodologies. Most people will use some object-oriented technique with UML.

4.4 High Level Design

This section describes in further detail elements discussed in the Architecture. High-level designs are most effective if they attempt to model groups of system elements from a number of different views. Typical viewpoints are:

1. Conceptual or Logical: This view shows the logical functional elements of the system. Each component represents a similar grouping of functionality. For UML, this would be a component diagram or a package diagram.
2. Process: this view is the runtime view of the system. The components are threads or processes or distributed applications. In UML, this would be a process interaction diagram.
3. Physical: this view is for distributed systems. The components are physical processors that have parts of the system running on them. For UML, this would be a deployment diagram.
4. Module: this view is for project management and code organization. The components are typically files or directories. This picture shows how the directory structure of the build and development environment will be designed.
5. Security: this view typically focuses on the components that cooperate to provide security features of the system. It is often a subset of the Conceptual view.

4.5 Low Level Design

This section provides low-level design descriptions that directly support construction of modules. Normally this section would be split into separate documents for different areas of the design. For each component we now need to break it down into its fundamental units or modules. For an OO implementation in Java, our components would become packages. Then the low level design will take each package and break it down into its classes. For smaller systems, you may have a single UML class diagram that each module description refers to.

4.6 Database Design

The section should reveal the final design of all database management system (DBMS) files and the non-DBMS files associated with the system under development. Provide a comprehensive data dictionary showing data element name, type, length, source, validation rules, maintenance (create, read, update, delete capability), data stores, outputs, aliases, and description.

4.7 GUI Design

The GUI of the system requires minimum user activity in order to get the desired output. The user is required to give an unknown image to the system as input, which the system is

expected to geolocate. However, for diversity and ease of use multiple ways of giving input image have been designed. For instance in desktop application there are three different ways of giving input image to the system

- Select image from pre-defined images in the given image slider for testing purposes.
- Select image from local machine through browse option.
- Capture and upload image from an android device directly and remotely through wireless connection.

In addition to all that the interface of the system has been designed with a particular ambiance so as to make it more understandable and promote the functionality of the system through the overall look of the system. The following figure of the system front end illustrates the point as it can be seen that the relevant graphical objects have been introduced in the interface to give a clue about the software.

Figure 4.3: Front End screenshot of the desktop application.

Chapter 5

System Implementation

As the design of the whole system has been discussed in detail in the previous chapter, this chapter focuses on the actual implementation details and explains how the design has been turned into a working system. All the techniques, algorithms tools and platforms will be covered in detail in this chapter.

The actual system has been divided into two major modules as described in 5.1 here both modules will be separately explained in detail.

5.1 System Architecture

5.1.1 Flickr Image downloader

The Flickr downloader is a desktop application designed and implemented in windows form application environment of Microsoft visual studio using C-sharp coding language. It can be further discussed in two separate module, The downloading module which actually interacts directly with the Flickr api [api documentation] made publicly available by Flickr.com for use in research and development endeavors. To be able to communicate with the Flickr API and request for public data, users must be registered with Flickr account and must apply for a unique Private API key[key link] this ensures that the data being requested is going to be purely used for development and research purposes.

The Flickr API is very mature and generating request links is very easy using the available interface online. Once the link is obtained with all the required fields it can be used in application to get image data for different tags with different input restraints. The http link upon request returns data in different formats including XML and JSON formats. This JSON response can be used to access actual images stored in Flickr servers. In order to parse through the JSON response appropriate classes must be defined to retrieve image data which is quiet complex procedure. For this purpose a very handy online tool[JSON to

C-sharp] was found and used which takes in JSON and generates classes to parse JSON response in C-sharp. This makes retrieving data from JSON response very easy. After the required classes were added to the source, HTTP request class available in visual studio was used to send request for data and obtain JSON response this response is deserialized to retrieve data image. The images are extracted from this data using those JSON classes and are then stored on the directory specified by the admin.

The second module is developed to upload the computed features of each image along with other necessary information to the cloud database. After the images have been downloaded the admin is able to update the database through this module. In order to make system and database more efficient and less costly in terms of storage, images are not directly uploaded to the database. Instead, the GIST feature of each image is computed and uploaded to the database along with its unique id, GPS co-ordinates and some other geographical information. For this to work programs utilizes the same WCF service used in all the other components of this project which takes in an image as a byte array and then maps an image from this byte array and initializes Matlab session for feature calculation the details of this WCF service are elaborately discussed in section 5.1.4.

5.1.2 Main application Front End

5.1.2.1 Webpage:

For the main module of this project which is the front end for the users to upload their query images and get results, asp.NET platform has been used to develop a web application.

The web front is designed to be user friendly and requires user to upload the query image to be assessed. Once the image is uploaded it's feature is calculated once again using the same WCF service which creates Matlab instances this service then return the computed feature of the image back to the asp.NET web application and from there comparison phase starts. The web application after receiving the calculated features connects to the SQL Database in cloud provided by Microsoft through its Azure cloud service. The application collects all the feature data from the database and stores in a variable to be compared to the query image feature the similarity is measured using Euclidian distance and the match is found by recording the minimum distance. After determining the minimum distance entry, the GPS coordinates are fetched from the database of the matched image feature. These coordinates are then mapped on to the google map extension shown on the web page front using google map and JavaScript functions. Addition information about the place is also shown such as city, country, capital etc.

5.1.2.2 Desktop Application:

In addition to Webpage, a desktop application has also been designed for the users to be able to keep on their local machines. The interface of the desktop application is pretty much consistent with the interface of the web application in terms of theme and ambiance. However the desktop application does have some added features for the ease of use. For instance the desktop application allows user to check location of an image in three separate ways. The first way is limited to test data set which is predefined and is in the form of picture slider to test the system the images of some location are added in this the slider and user can select one of these images to check if the system is functioning properly. Another way in which the user can provide its own query image by browse option through which the users upload the image from their local storage and this application then predicts the location of the image. The most unique and exciting way of providing image to the application is by using a remote android device connected to the user machine via Wifi technology, this allows users to capture an image on their local machine using their android device with live feed streaming to the desktop application.

The process after is quiet exactly similar to the web application module Once the image is uploaded its feature is calculated once again using the same WCF service which creates Matlab instances. This service then return the computed feature of the image back to the desktop aplication and from there comparison phase starts. The application after receiving the calculated features connects to the SQL Database in cloud. The application gets all the feature vectors from the database and stores in a variable to be compared to the query image feature the similarity is measured using Euclidian distance and the match is found by recording the minimum distance. After determining the minimum distance entry, the GPS coordinates are fetched from the database of the matched image feature. These coordinates are then mapped on to the google map extension shown on the browser control which is embedded on the windows form application and is activated once the results are computed. The map is shown on this browser control using same html page used in web application, utilizing google map and JavaScript functions[7]. Additional information about the place is also shown such as city, country, capital, population and name of place.

5.1.3 GIST Descriptor:

The most valuable image feature used in this project is the GIST descriptor [8]. The GIST Descriptor is a high level powerful image feature which evaluates the gradient information of the image and is used to understand the low dimensional representation of the scene. It is in most cases used in scene matching approaches which require understanding the overall context of the images. It is generally used to understand the context of the image without having to identify individual objects in the given image using object recognition.

5.1.3.1 Gabor Filter:

The primary component used in this regard is Gabor filter. The basic idea is to convolution on the given image with Gabor filter window of 8 different orientations (angles) and 4 different scales (lengths). This results in 32 different feature maps and each map is then divided into a 4x4 grid. The average values are computed for each bin or region of grid. Combining all these values results in a single vector of length 512 (8x4x4x4).

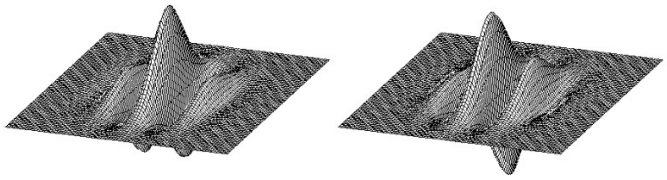

Figure 5.1: Graphical representation of impulse response of a Gabor filter

The above mentioned description is the general idea of the GIST descriptor, the actual implementation has seen many variations depending on the application and context of use. For this project the appraoch used to compute GIST of an image is spatial envelope based computational model for recognition of real world scenes, proposed by Aude Oliva [9].

5.1.4 WCF Service

The WCF (Windows Communication Foundation) [10] is service-oriented application developing framework. These kind of applications do not directly communicate with the users but can provide services to the applications interacted by the users[11].

For this project the WCF service was developed to provide common service point for different kind of applications. For instance the front end web application and image downloader desktop application both use the same service. Furthermore through the use of service it could be made easier to expand the scope of the project while practicing reusability and develop application on other platforms consuming the same service, like mobile based android applications.

The WCF service basically contains the core method written in C-sharp which creates and Matlab instance and calls Matlab object using COM object and Matlab library in visual studio. As already mentioned in earlier chapters that the integration between Matlab and other development platforms is quite difficult and limited due to strict Matlab policies. Still some basic procedures are available which allow using Matlab on other platforms. The procedure used in this project does not allow sending complex datatype like image or even byte streams to Matlab function so in order to provide Matlab function with the image a simple technique was used. Each time the method in WCF is invoked the application utilizing service must send an image by converting it to byte stream for easier communication. Once the byte stream is received by service side method it is reconstructed

into the bitmap image. This bitmap image is then saved onto the disc along with a text file for the output this process is done using a random number as the name of the image, generated each time an image is saved. The text file is also saved with the same name as the image and the paths of these two are passed to the Matlab function called using Matlab instance. The Matlab compiler starts up and the called function is executed. The function picks the image from the path passed to it by the service and computes features, the result is then stored in the text file using the path also provided by the service.

As the function executes completely the code returns to the service and the Matlab instance is closed. The data is then read from the text file and stored in an array which is returned to the service calling application. Also the picture and text file are deleted from the disk at the end of the service code in attempt to clear memory and avoid unnecessary memory consumption.

5.2 Without service

For the ease of demonstration and in some cases to limit the use of resources a version of desktop application is also developed which runs independent of the wcf service. This is strictly for demonstration purpose and this version of the application directly creates the instances of the Matlab and gets data from it. In some cases this might be more feasible than having an intermediate connection where data is sent and received indirectly. But for some cases it also better to have a common point which when changed or updated or edited, automatically applies to the whole system and changes are needed at only one point, that's the long term benefit of the WCF service that it provides a common ground and consistency when changes are made.

Chapter 6

System Testing and Evaluation

6.1 Graphical user interface testing

The user interface of the system is quiet simple and self explanatory the user is just required to just give input image and the system shows the output by locating that picture on the map. So the testing in terms of GUI is not necessary.

6.2 Performance testing

Considering the performance factor the system does take some time to process the information provided which can also be due to the use of not very powerful machine which acts as both client and server while running the web application. Also due to the distributed nature of the system in which an instance of Matlab is also generated the system tends to take some time. Infact the Timeout limit for the system had to be increased to avoid timeout exception when the system takes more than a minute to respond.

6.3 Functionality testing

In terms of functionality different inputs have been provided and output has been observed. The system in most cases returns correct output i.e. in more than 90 percent cases the system predicts the right class or location of the control input. Though in about 10 percent cases the systems does give wrong output. This of course can be reduced by adding more data as consistently mentioned in previous chapters. Following are some test cases designed to ensure error free system running.

Test case ID	**TC - 01**	
Test Description	Testing picture scroll	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Click Scroll button	Pass
2.	Scroll Left	Pass
3.	Scroll Right	Pass

Table 6.1: Test Case: 01

Test case ID	**TC - 02**	
Test Description	Testing picture button	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Picture Selection	Pass
2.	Picture shown in selected picture panel	Pass

Table 6.2: Test Case: 02

Test case ID	**TC - 03**	
Test Description	Testing Load control	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Load form opening	Pass
2.	Loading icon shown on form	Pass
3.	Lock main form until load complete	Pass

Table 6.3: Test Case: 03

Test case ID	**TC - 04**	
Test Description	Loading data from database	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Fetch data from database	Pass
2.	Storing data in variable for comparison	Pass

Table 6.4: Test Case: 04

Test case ID	**TC - 05**	
Test Description	Sending Test Image to Matlab	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Save image on disk	Pass
2.	Get image Path	Pass
3.	Create Matlab Instance	Pass
4.	Send image path to Matlab	Pass

Table 6.5: Test Case: 05

Test case ID	**TC - 06**	
Test Description	Compute Image Feature	
Initial Condition	Matlab Application is Running	
Step	Task Performed	Result/Evaluation
1.	Read image from disk	Pass
2.	Extract Image Feature	Pass
3.	Save Image feature	Pass
4.	Return to Main Application	Pass

Table 6.6: Test Case: 06

Test case ID	**TC - 07**	
Test Description	Compare Image features	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Get image feature computed	Pass
2.	Compare with database image features	Pass
3.	Determine nearest neighbor using Euclidean distance	Pass

Table 6.7: Test Case: 07

Test case ID	**TC - 08**	
Test Description	Show image location on Map and geographic details of location	
Initial Condition	Application is Running and Internet Access is available	
Step	Task Performed	Result
1.	Get GPS data and geographic info of nearest neighbor from database	Pass
2.	Show detail (City, Country, Capital, Population) on label	Pass
3.	Load Html panel for map	Pass
4.	Get google map from API	Pass
5.	Plot GPS coordinates on map	Pass

Table 6.8: Test Case: 08

Test case ID	**TC - 09**	
Test Description	Interact with google map	
Initial Condition	Application is Running and Internet Access is available	
Step	Task Performed	Result/Evaluation
1.	Load google map	Pass
2.	Zoom in on map	Pass
3.	Zoom out on map from API	Pass
4.	Click on pinned location to show pop-up detail	Pass

Table 6.9: Test Case: 09

Test case ID	**TC - 10**	
Test Description	Select image from local directory	
Initial Condition	Application is Running	
Step	Task Performed	Result/Evaluation
1.	Click browse button to invoke open file dialog	Pass
2.	Select image file from local directory	Pass
3.	Get image file path to textbox	Pass
4.	Click Upload Image Button	Pass
5.	Get image location from textbox	Pass
6.	Read image from given directory	Pass
7.	Show image in selected image panel	Pass

Table 6.10: Test Case: 10

Test case ID	**TC - 11**	
Test Description	Get image from Android device	
Initial Condition	Application is Running, Connected to Android Device via WiFi	
Step	Task Performed	Result/Evaluation
1.	Connect to android device via hotspot	Pass
2.	Open android module	Pass
3.	Open new form with live android camera feed	Pass
4.	Capture image	Pass
5.	Load image to main form	Pass

Table 6.11: Test Case: 11

Chapter 7

Conclusions

The most useful experiences which lead to learning of completely new concepts and techniques are as follow:

- The use of API and interaction with API resulted in learning of alot new concepts such as using XML or JSON to retrieve data from web applications. This also helped better understanding of how API's actually function and how requests and responses work.
- Serialization and Deserialization of objects was also implemented in this project which was a new concept and efficient way of sending and receiving data.
- Integration of Matlab with C-sharp was also very difficult and lead to learning of alot of new concepts like COM objects and Matlab Runtime Compiler for generating executable files.
- As cloud service was also used in this project so the implementation and usage of cloud database was also learnt.
- In the main module JavaScript was used to map the input image using google maps API and extension. Proper use JavaScript was also a learn able experience.

References

[1] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. Cited on pp. 1 and 5.

[2] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–6. IEEE, 2007. Cited on p. 5.

[3] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000. Cited on p. 6.

[4] Dr. Neal Krawetz. "tools, techniques, and tangents". Cited on p. 7.

[5] Guang-qiang YAO and Li-ping CHEN. Integrated programming between c# and matlab based on component object model [j]. *Computer Engineering*, 14:032, 2008. Cited on p. 15.

[6] Shi-wei Zhao, Ming-bo Zhao, and Ping Chen. Implementation and application of matlab & c# .net integrated programming based on com [j]. *Journal of Shandong University of Technology (Science and Technology)*, 4(007), 2006. Cited on p. 15.

[7] Bing Pan, John C Crotts, and Brian Muller. Developing web-based tourist information tools using google map. *Information and Communication Technologies in Tourism 2007*, pages 503–512, 2007. Cited on p. 20.

[8] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 19. ACM, 2009. Cited on p. 20.

[9] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001. Cited on p. 21.

[10] Alex Mackey. Windows communication foundation. In *Introducing. NET 4.0*, pages 159–173. Springer, 2010. Cited on p. 21.

[11] Markus Stopper and Bernd Gastermann. Service-oriented communication concept based on wcf .net for industrial applications. In *International multi conference of engineers and computer scientists*. Citeseer, 2010. Cited on p. 21.