ABSTRACT

As the number of images on the internet resources have grown enormously to a large extent in the past few years, the development of efficient techniques for the purpose of navigation, identifying, indexing, searching, and labeling the images has become very important.

In this research, we will rely on two major resources and features for the identification of images, one is the image features while the other is the text based tags attached to the images on photo sharing sites. The image based search is done by the extraction of certain high and low level features from the input images and all the images in the database, and are compared as per the different machine learning algorithms to compute and show the images most nearer to the input image. Different types of features are extracted from the images with the help of basic image processing techniques including histograms, scale values, and other relevant features. The tag and text based searching is done by the help of comparing the tags attached to each of the images with the tags that are input with the help of machine learning and artificial intelligence algorithms to find out the nearest tags in the database. Certain tagged are considered noisy and are eradicated, the preferences and importance factors plus distance are attached with the tags, and the number of the tags plus the number of exactly matching or closely matching tags are also kept in consideration for the purpose of short listing the relatively close images.

The project takes use of the online photo sharing resource on Flickr due to two different reasons. It is one of the biggest resources available with a large number of pictures, and some work has been done for the creation and development of third party application programming interfaces for the extraction and comparison of images from Flickr. This helps in the extraction of images on the basis of certain particular tags and saving them in the database so that the algorithms can all be implemented later on the information stored in the database. In this project, we extract around twenty thousand images from Flickr on the basis of four different tags, and those images along with the tags and all other relevant information are stored in the database. Then the certain image features of each of the images are extracted and are saved in the database as well for the purpose of matching later on. Then whenever a text or an image is entered, the database is queried on the basis of certain efficient and effective machine learning algorithms to find the appropriate close results with the help of attaching weights and distance factors which each kind of query. Different effective and highly researched algorithms like similar text searching, k-nearest neighbor, and neural networks are utilized to find out the closest matches, and then the closely matched images are displayed as an output to the application on an effective and usefully designed user interface on .Net platform.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background

As the digital cameras and the use of internet technology has evolved and increased to an unbelievable extent in the past couple of decades in everyday use, the number of images available on different sources around the internet or on personal databases has grown to an enormous number as well. This can be understood by sighting the statistics on some of the famous platforms for online photo sharing. Flickr, a prominent resource and website for the storage of photographs (Amazon), consists of over 6 billion images (Flickr). There is a lot of effort and a huge amount of database servers and bandwidth required to handle and manage this type of excessively large databases of photo sharing, but at the same time there are many advantages attached to the presence of such systems with the availability of a very large number of pictures. These large numbers of photos available on the online resources leave room for many new possibilities and applications by investing in the field of statistical analysis, image processing, automatic model learning of images, and machine learning for the classification, identification, and indexing of the images.

Currently the images are searched by most of the available on the basis of some tags and some information related to the camera and time of picture taken that is sometimes available with the images and sometimes not. The biggest problem associated with the searching of images on the online resources on the basis of tags is that many people tend to highlight their own opinions and interpretations while uploading or attaching tags to the images which introduces a need to implement proper machine learning algorithms to find matching texts and to get rid of the unimportant tags that are not accurately needed. For example, when a tag named as Christmas, is searched along the internet on the photo sharing sites, the most common and important output is a large number of pictures that have almost nothing to do with the event Christmas or its exact calibrations. At the same time, the kind of texts to be searched have to follow certain criteria because number of pictures can be found on the data 25th of December regarding Christmas than by the work Christmas itself. That is one reason that depicts accurately that the searching of the images solely on the basis of text tags is not enough in any way at all, and there is a need to make use of more than one type of features for the proper and effective searching of the images.

In this particular work, the focus has been shifted from the normal implementation of such system solely on the basis of text to the searching with the help of a couple of methods i.e. text based searching from tags and image based searching for the features. A major effective future application of this type of project can be considered to use a

combination of both of the methods to detect similar images on the online databases which can be much more effective and free of noise.

The focus in terms of the online repository is kept on Flickr. Flickr is one of the largest online photos sharing websites and it contains over 3 Billion online photos shared from different people from all over the world. This is one of the reasons why third party APIs have been developed to assist in the process of searching images through Flickr on the bases of certain texts. The saving of images is important in the regard that they can be saved with all their tags on the databases that are further utilized for the purpose of searching and implementing machine learning algorithms. There are many other advantages also attached with the use of Flickr as compared to the other resources. It provides a huge amount of training data that is annotated, but the annotations are often quite noisy and unimportant, hence introducing a need to filter appropriate data.

There are a large number of applications in which this particular research and implementation can help people. One of the main advantage is in the domain of tourism where people can upload a picture where they are and can see pictures of many related images to that, and they can upload the tags regarding the place they are in and can know about all the places that exist in the region. Apart from this basic application, it can also largely help in the improvement of the GPS based applications and the help of such applications can be taken by the investigation departments. The research also leaves a lot of room for further conduction of research in the field domains of image processing, databases, machine learning, and artificial intelligence. So this project can be taken for further improvements to develop very important and helpful future systems.

1.2 Motivation

The internet has become a repository for the collection of several important things like photos in the recent times. Recent estimates show that the number of online photos available at present is in the range of tens of billions which is actually a very huge amount. The largest image warehouses among the online resources are Facebook, Flickr, ImageShack, and several others, and while some of the platforms like Facebook give the option to also contain the private pictures which are not visible to the rest of the people, some platforms like Flickr have all of their images open to be viewed and accessed by anyone around the globe. The number of freely available images on Flickr ranges from over 3 billion pictures to 5 billion pictures in total which is actually a very huge number (Eric, 2009). This is the first time in the history that someone has the access to this large number of pictures.

These photos that are present on these open resources are not limited to certain places, certain countries, or certain people, rather the pictures are available from all around the world. So there are pictures available from pretty much every country and every city in the world which makes the application of such a type of research resourceful and helpful for people residing in any part of the world. To highlight some of the statistics, the search for the city ‘Rome’ has around 2 million photos on Flickr, 3.5 million photos of Tokyo are available, 3 million from the city ‘Seattle’, and same is the case for almost any city around the globe.

The purpose of showing the details of all of these cities, and the purpose of providing all of these statistics is mainly to highlight that the number of images for each and every location in the world is quite huge, and that is a major advantage for this particular project as this application can be used by anyone anywhere in the world for achieving the purposes of GPS as well as for the purposes of tourism.

1.3 Scope of the research

In this project, different web techniques are under consideration to create a local database. The tag clouds are used as a data source to discover the relationships between of different tags. GPS information, user id and image description along with tag cloud is saved in the database. Image recognition techniques are also focused to match user fed images with different images saved in the database. Flickr website will be used as the primary data source for all the images to be saved in the database. There are a large number of fields and domains involved in the conduction of the research for the topic under consideration and for the implementation of the project. The topic takes within itself the domains of image processing, machine learning, artificial intelligence as well as database management and web based retrieval. So, there are many different fields within the domain of this particular project.

Image processing

The field of image processing lies in the domain of signal processing in which the input to a system is an image on which different algorithms are applied and then the received output can be an image, a feature, or a set of different features. In our case, the input to the image processing system is an image while the output is a set of features after the implementation of different algorithms.

Artificial intelligence

Artificial intelligence is basically a field or domain in computer science that deals and aims at the creation of intelligent machines that can perform certain functionality after consideration testing. In our application, the program is designed after training and modifying for a large number of inputs, and now the algorithm has been developed in an artificially intelligent manner to handle a large number of unsupervised testing samples.

Machine learning

Machine learning basically lies within the domain of artificial intelligence that deals with the training of a set of input data for the training of the program before actually implementing it for the testing phase. The algorithms related to machine learning in this particular research topic and implementation are k-nearest neighbor, neural networks, and matching text finding algorithms.

Database management

Database management lies in the domain of computer science and software engineering for the purpose of maintaining data for easy and quick retrieval and updating. The data is kept in another program in the database with all the images, tags, and image features plus other relevant information, and all the algorithms are implemented with the help of queries to the database.

1.4 Aim of the project

The aim of this project is to optimize the image searching techniques based on tag clustering with most of the images on Flickr website. Using machine learning the relationship between different images can easily be furnished which can make the tag based search more easily and much optimized. Different machine learning approaches can be utilized to achieve the desired targets. On Flickr, pictures can be searched using a keyword. The resultant picture contains lots of other information other than the tags e.g. location information, user who uploaded the image and text describing the image in a few sentences. The project also aims to develop relationships between images using basic digital image processing techniques on the images collected from the Flickr website.

1.5 Research design

Introduction

This chapter contains a brief introduction about the research topic, the purpose of the research and key questions and topics to consider for the research.

Literature review

This chapter critically reviews the work and research previously done on the dissertation topic. This chapter explains all the related topics from different literature resources. It also includes conclusions and findings from the previous research done on the topic. This chapter does not contain any opinion or no topic of the research is explained here. It only contains the previous research done by other people on the topic and analysis on that research explaining why or why not the under consideration research can be implemented in the current topic.

Research methodology

The research strategy, research methodology and data collection techniques and methods for the research and mentioned in this chapter. The method by which the outputs of the research are achieved is briefly explained in this chapter. This also contains the motivation behind the research and research ethics topic is also included in the chapter. It also contains basic questions to be considered about the research.

Application Design

This chapter includes the research done on the topic. Computer application which is designed for the research is explained in this chapter. The tools used to design for the programming of application and third party API's are explained in detail in the chapter. The programming techniques used in the project are also explained in this chapter. The algorithms used to fetch images from flickr.com and searching for images on the basis of tags online from the flickr.com is demonstrated in this chapter. Database design and offline image search for the images stored in the database is also explained in this chapter.

Machine learning algorithm and the image processing part are also explained with complete explanation of the code is explained in this chapter.

Results

This chapter contains screen shots of application outputs. Step by Step working of the application with GUI form explanation is done in this chapter. The accuracy and calculations of the results of the implemented software are displayed and discussed in detail in this section. Desired results are compared with the expected results to check the accuracy of the research.

References

Complete list of references used in the dissertation is mentioned in this chapter.

2 BACKGROUND

2.1 Introduction

The purpose of this chapter is to review the work and research done previously on the related topic by other people. Different data and information on the implementation of tag based image search algorithms and their optimization is studied in this chapter. The purpose of the literature review is to gain in depth knowledge about the topic before actually designing the implementation of the project. All related previous data is studied in this chapter and no analysis or views are given on the research in this chapter.

2.2 Flickr

Flickr is one the most popular online repository of images where people can upload pictures which can be viewed by anyone on the website based on the related search. The Flickr website asks the users to add related keywords while uploading an image to the website. The keywords must be such that the images can easily be recognized by the keywords. Although there is no limitation on the tags quality and quantity but Flickr encourages their users to add as many keywords that can be related to image while they are uploading an image, adding unrelated tags to the website with an image can only increase the load of data on the website server and cannot be used effectively by the website. Flickr does not limit their users to add only a specific set of images but the images can be about a place, person, event, people and the images can also be the personal pictures of the users uploading the images on Flickr website. Today, Flickr contains more than 6 billion pictures on their website.

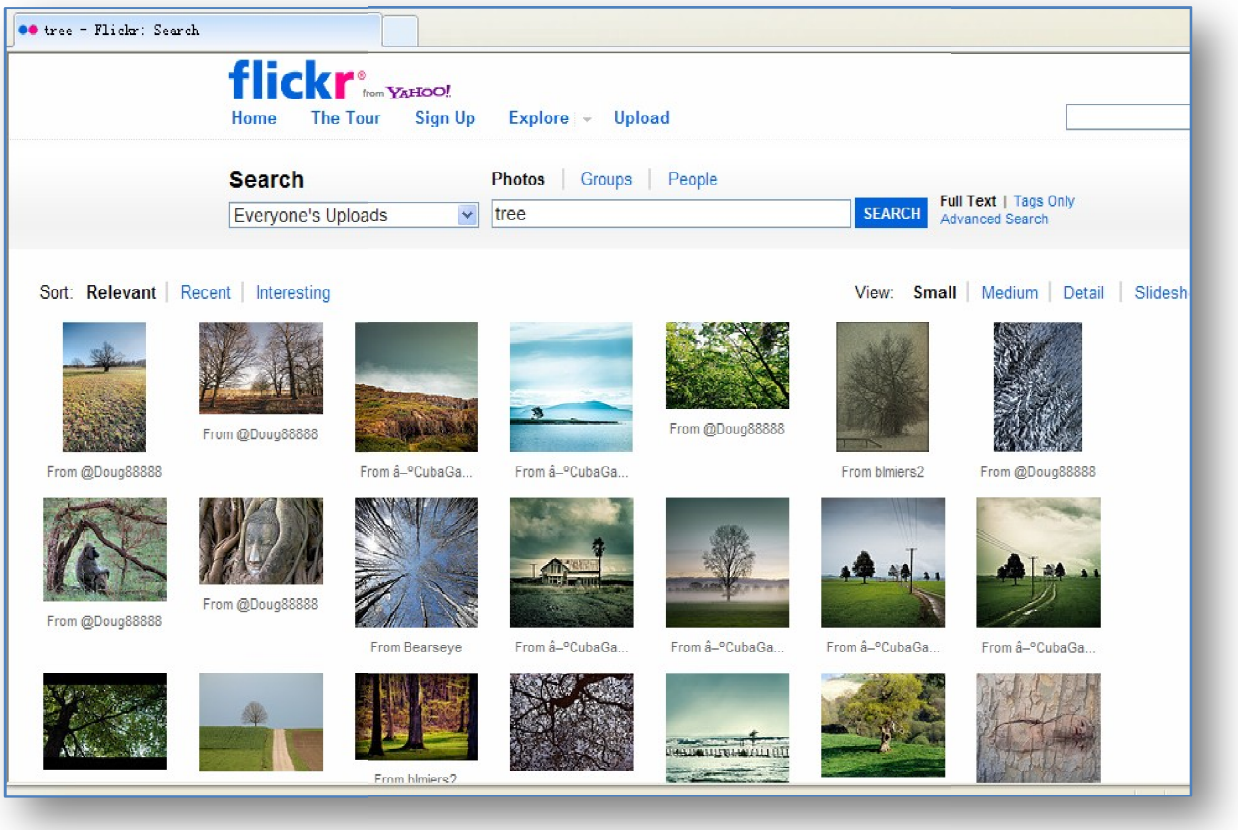


Figure 1. The website of fliker

The tag cluster system implemented by Flickr makes it easy to connect different images uploaded by different people and thus the tag based system are a key success factor of the Flickr website. A tag is not a collection of sentences but it contains a single word that can be used to describe the image, tag cloud on the Flickr website does not contain only 1 or 2 tags but it may contain as many tags as required by the user. It not only helps the Flickr website to easily connect related images but it also helps the users to organize their images on the website in a much easier way. Tag system is not implemented only by the Flickr website but over the past few years it became very much popular in many other social sharing websites. The tag system can not only be used to describe images but many of the bloggers are also using this system to describe text based blogs which makes the search process on their website easier.

2.3 Tag systems

Tag system is very much common today in the online world especially in the social networking context. There are different methods and techniques used by different people to organize and manage tags according to their requirements. These tags help other people to view information and resources available on different websites. All the sharing

patterns and folksonomy completely relies on the tags describing the relative data. Folksonomy is a classification system which is derived from collaborative tagging and categorizing different tags. In sharing prospects on the internet on websites there are usually two stakeholders involved. One is generally referred as the producer who is actually uploading information on the internet and then there are the consumers who are utilizing that information for the desired purpose. The producer plays an active role which sends the resources to the users according to their requirements and preferences gathered generally from the user browsing history. All the resources available on the internet are generally categorized with tags. Some visual technique such tag clouds, recommended list can be used to assist the producers. The producer can publish resources via different web sharing methods like sharing of pictures or sharing of text or documents. Information retrieval is that users actively search for the resources by mentioning different tags related to their requirements. The consumers can find results which match their requirements based on the tags provided by the user. The tags searched by the user can also give suggestions for a future tagging process to the producers. Thus users contribute resources to the website which makes the website to act as the producer but websites are controlled by other people and resources are always from other people making websites only a medium to provide a platform for the whole sharing process. Tags are usually a short text that depicts the content of the picture. For one picture, there could be multiple tags associated with. Based on the information provided by tag itself, there are two categories of tags. A good tag is a tag which defines a unique, specific and accurate description to one aspect of the picture. There is no ambiguity of this tag. A bad tag is a tag provides less information about the picture. It could be a common word, unrelated description or even misleading. A good tag can reduce the search time and provides an accurate description of the picture. We need to reduce the number of tags which cost more time to discover and recognise.

The nature of tagging system allows user to choose arbitrary words for their resources. However, if the user lacks of related knowledge of the resources, they may not choose the accurate vocabulary to describe the resources. The predefined taxonomic structure for the concept of resource may not be popular amongst all the users who can tag the resource. For example, a picture which depicts the Big Ben could be tagged with Bell Tower, tall building, building in London or landmark of Westminster. The tags are not the accurate description for the picture, thus it causes the trouble for tagging system.

The result of unaware tagging, that is, the tags which are redundant, inaccurate, mistake, is undesirable for tagging system. The disagreement of vocabulary of the tags will introduce the inefficiency of the tagging system. The user interaction of the same resource is also affected.

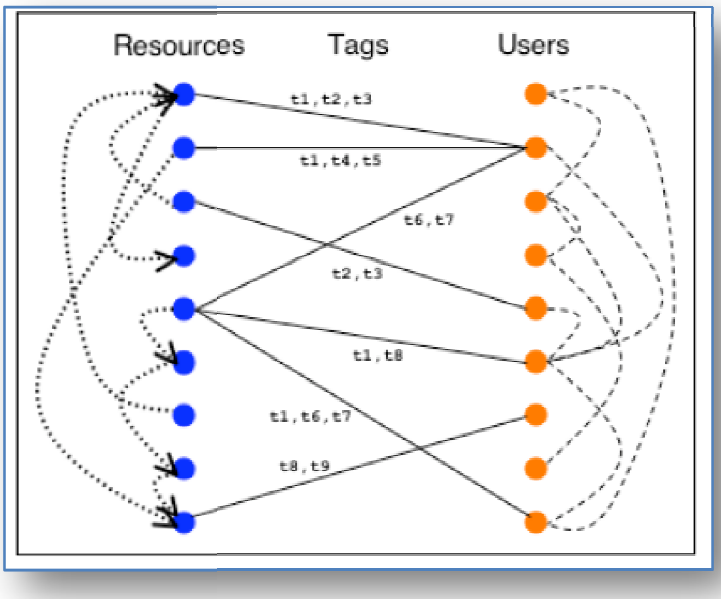


Figure 2.the concept model for tagging system

The concept model (Figure) consists of three parties. The user can create a tag and assign it to a specific resource. The tags are the edge of the connection between users and resources. The resources can connect to each other (for example links connect the web pages, but we do not use these relations in our problem, i.e. the pictures). The users can connect to each other as well (for example in social network, the users may have the relationship between each other. We ignore this connection too for the sake of the simplicity of our problem).

2.3.1 Tagging in social networking

Tagging allows the user to choose different words for their resources arbitrarily. Thus if the user lacks the related knowledge about the data to be uploaded they may not choose the correct tags or tags may have vocabulary mistakes and thus the whole tag system can fail to work if it does not follow some specified standards. The predefined taxonomic structure for the concept of resource may not be popular amongst all the users who can tag the resource. For example, a picture which depicts the Big Ben could be tagged with Bell Tower, tall building, building in London or landmark of Westminster. The tags are not the accurate description for the picture, thus troubles the complete tagging system. The result of unaware tagging, that is, the tags which are redundant, inaccurate, mistake, is undesirable for tagging system. The disagreement of the vocabulary of the tags will introduce the inefficiency of the tagging system. The user interaction of the same resource is also affected. The conceptual model of the tagging system contains. The user can create a tag and assign it to a specific resource. The tags are the edge of the connection between users and resources. The resources can connect to each other based on these tags which make the user interaction lot easier. The contribution of the users is mostly tags for resources. The study around the connection of resources is mostly about

web pages. The social networking study is focusing on the relationship between the users. It is helpful to apply the measurement development in social networking to tagging system. If all the tags that are used to tag the pictures, the users may not continue to add the synonyms of available tags to a specific picture. In some applications, the users are not aware of the tags created by another user. The chance of appearance of redundant tags can increase greatly. The links between the users are crucial to social network. However, the influence of functions of social network may make the tags system in picture sharing easier. The users may provide some synonyms of the tags that could provide us clue of the relationship between the tags. In this scenario, the tags are actually bond to the resources. The resource became the bridge of the tags which are describing the attribute of the resource. We can connect all the tags which may share the same meanings to a certain resource. Taxonomies for tagging in social sharing network may be categorized into two basic things. These factors involved in the social tagging system may be the producers or the consumer. The producer plays the role of controlling the tags about a certain resource. The methods of distributing the tags many restrict the availability of information. The users or consumers can promote the selection of both resources and tags. The contribution of the users is the source of the tags. If we provide enough incentives, the users are willing to provide enough tags for the resources. The user assistant functions are one form of incentives. The difficulties in tagging system, as pointed out by Golder and Huberman (2007) is two aspects of semantics. The poly-semi is one single word but it can have multiple meanings. The same word with different meanings is very hard to detect. Supplied by a dictionary, the word with different meanings can be detected. However, it is difficult to select the correct meaning for the word. In some occasions, the word can be used for multiple meaning at the same time. The synonymy is multiple words having the same meaning. The synonymy is relatively easy to detect and tackle. In addition, the ontology of word that the various level of abstraction of the word in the same tag list can be difficult to recognize and categorize. According to Marlow (2006), the tagging system can reduce any number of tags into two-dimensional taxonomy. The dimension of taxonomy is the way to study a snapshot of tagging system. The first dimension includes two aspects, tag creator and resource creator. The second dimension is based on the aspects of first dimension.

2.3.2 Taxonomy Factors

The factors that can improve the quality of tags and the complete tagging system can be categorized as taxonomy factors. To enhance the functionality of tagging system there must be some rules and regulations for both the producer and the users or consumers. Both the producer and the consumers have some responsibilities to make the system more effective.

The shared tags are a tool to attract users to browse the resources which have the same or similar tags. To view those related tags, or explore the related content via tag clouds, user can engage the activities for play or compete with the counterparts. The incentive within those social activities provides user a way to present and express their identity. For example, the users can express their habit, personality, preference or track of life by publishing and tagging their resources on the online service. This is the incentives and motivations based on the human nature. The basic incentives and motivation embedded in the nature of online service is to provide sharing services for users to easily organize and

publish their resources. The two motivations, organizational and social, are the basics amongst all the motivations. These incentives and motivation drive the users to tag the resources.

These taxonomies as mentioned by Trattner, Korner and Helic (2007) are as follows, producers must have to follow the following basic rules which may include access to tags is a restriction on what user can see about tags. There are two modes for users to access the tags. As in dimension, the users can either have full access to the others' tag or simply partial access. There are some existing models adopted by online service. Technorati.com a famous technology blog allows users only to tag the resource of their creation. Yahoo allows users tag any resources, but allow no one remove the tags. Granting users have more access to tags could reduce the noise in the tags list of resources. Tag hint is a method to provide supplement information to assist users pick up a tag. The tag hint is not always available in online service. Yahoo provides the tag hint system as a viewing existing tags and suggestion list. Delicious bookmark did not provide tag hint. In tagging hint system, usually users can select from a set of relevance tags according to the existing set of tags. The hint may also be extracted from the contextual metadata, the extension of the existing tags based on recommendation list, or the tags from similar pictures. The multiple hints guide the users to pick up the tags which are convergence in folksonomy.

However, the suggestion of tags for a certain resource could promote a certain tag unfairly. The most popular tag is the one most users select. There is a chance that a tag is overweighed. Connectivity of resource is the internal graph represents the relationship between the resources. The organization paradigm of the resources can affect the tagging system as well. For example, in Flickr, the pictures can be grouped under a certain group. The group is an important factors that affecting the tags. If the group is the name of a place, we have valuable implications for the tags which attached the pictures within the group. Connectivity of User can be useful to discover the connectivity of the resources. The social connectivity can be a factor for statistics for calculating usage of the tag hint.

The responsibilities of the consumers may include many of responsibilities like marking the resources storage which is provided by many of the websites. Mark the resource as potential member of a cluster or group is helpful for reminder to users. Users can take advantage of the benefits of the reminder function. This storage assistance can be used for the resources that associated with no metadata. Social Features involve multiple user take activities on the same resource. The activities include sharing, competition, and attraction for unknown. The users can also comment to express their options. The shared tags are a tool to attract users to browse the resources which have the same or similar tags. To view those related tags, or explore the related content via tag clouds, user can engage the activities for play or compete with the counterparts. The incentive within those social activities provides user a way to present and express their identity.

For example, the users can express their habit, personality, preference or track of life by publishing and tagging their resources on the online service. This is the incentives and motivations based on the human nature. The basic incentives and motivation embedded in the nature of online service is to provide sharing services for users to easily organize and publish their resources. The two motivations, organizational and social, are the

basics amongst all the motivations. These incentives and motivation drive the users to tag the resources.

2.3.3 Tag cloud system

Tag clouds, or weighted cloud, are a visual representation of tags. The idea of tag clouds is to emphasis the most important tags and to understate the less important tags. The importance of a tag is based on the frequency of its appearance in all tags. Tag clouds are a way to stratify the tags. In a typical tag cloud depict, we can easily note the tags which are more popular. On Flickr, the tag clouds are sorted alphabetically. The weight of popularity of a tag is expressed by the font size of the tag name.

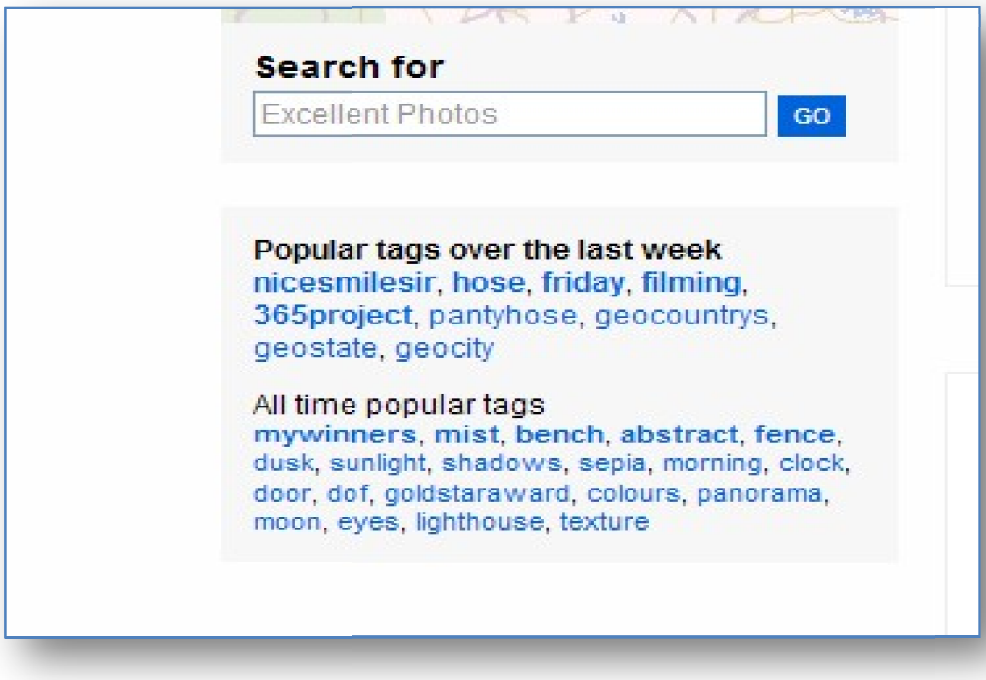


Figure 3.Tag cloud from flickr

The study of the tag clouds helps us to learn the selection of top tags around a theme, a topic and a picture. The work around tag clouds can also be used in recommendation list for a certain set of key words.

2.3.4 Tags on Flickr

Tags on Flickr are all user generated. Tags are a powerful and useful feature in many social media and Web applications. By using tags, the user can mark the pictures on the web the use of tags is widespread in other resources on the web e.g. the blog, bookmarks and video. Comparing to ontology system of categorizing resources, tags are free-style, no hierarchy in the organization. All tags are in the same position for marking the aspects of content of pictures. It eliminated the structure which is demanded in ontology and categorization. As a result, tags are easier to pick than categories from ontology. Due to its flexibility, tags are also evolving to reflect emergent properties of their referents as mentioned by Golders and Huberman (2006). Tags, which lack of priori structure information, contain less semantic meanings. To achieve the same usable result as categories, which are defined based on ontology and semantics; we need to discover the structured information from the tag-based system.

The geo-referenced tags on Flickr often contain the tags of the names of places and general location. The semantic analysis on place names to cluster the pictures which share the common word of places. By definition, the place semantics indicates the geo-locations and tags are strongly associated. The mapping from tags to geo-locations is the pattern that can easily be discovered. In place semantics, the definition of tags i.e. it can be used as a powerful data source. Pictures are defined as (id, user, latitude, longitude, description, location, web address etc).

2.3.5 Getting information from tags

Different sort of information can be gathered for tags using many of techniques currently available. These techniques may include the following:

Semantic analysis

To cluster tags based on their geo-location implications by intuition, for one geo location, the place/location tags will burst, that is, a certain number of the tags with the same name cluster in that region. This is the principle of discovering the tags for the certain region. Secondly the region with certain of characteristics often has a boundary. Either one characteristic can be extended to the regions nearby, or the characteristics diminished to a certain boundary. For one characteristic, the region which has these characteristics may not overlap. For example, the income distribution is different than disease distribution in a certain region. The last problem is to discover the correlation of the regions. Many

regions of one common characteristic could be neighbors in one area. We can use correlation tool to integrate small regions to a large one. In addition, another aspect of semantic analysis on tags is about the culture localization. Some places, like art or museum is more correlated to a place instead of an abstract word. In order to identify which semantic meaning of a tag could be used as place/location, or first-order phenomena. We consider the tags only for its meaning in general for simplicity. For one certain region, the frequency of tags appearance in the tag list of a picture can be detected by algorithm burst-detection introduced in signal processing. The rationale behind this assumption is the consistence between semantic meaning and spatial region area. In a certain region, people tend to describe the region by the similar words. It is like the high peak of a probability distribution. Time-series analysis technique is applicable to spatial area analysis as well. The general idea behind the common time-series analysis algorithm such as ARIMA which was presented by Box and Jenkins (1976) and McDowall (1980) can detect the burst not only in temporal domain but also in spatial domain. The results of spatial analysis are sensitive to the definition of spatial unit. To reduce the influence of distributed data in spatial dataset, it is common to normalize the dataset, making to related to population or population density. Thus, the result of spatial analysis is equalized with the variables, and it can be used more accurately.

2.3.6 Tag mapping method

The tag mapping method is specialized in identifying tags for the geo-location region automatically. Ideally, the tag mapping method will search for the tags that are uniquely define the geo-location region. For example, the tag mapping method can easily give the high score for the tags such as Big Ben, London or The London Eye. These unique names are the first choice of tag mapping method. The assumption of this method is that the unique name only specifies one place in all places all the time. The places which are different from current place must be named differently. The tag mapping method can learn the spatial region from the data, which is very different from the burst detection methods such as native scan and spatial scan. Due to the learning nature of tag mapping method some other ways must be found to define the scale and partition for the whole spatial domain. The burst detection methods demand the explicit ways to find the scale and partition for the spatial domain. The tag mapping method can learn the parameters of scale and partition from the picture data by cluster procedure. If the result of cluster is very small, the scale and partition is relative larger. The first step is to find the parameters of scale and partition. We can apply the common machine learning algorithm to find the parameters for one dataset. For scale parameter, we use the number of clusters generated from the photo data.

2.3.7 Scale-Structure Identification Method

Comparing to tag mapping method, the Scale-Structure Identification Method use the similar process to perform the significance test. The difference in scale-structure identification method the cluster is not used by its total number in a certain region, but the minimum spatial distance between clusters.

We make the assumption for place tags as follows: the spatial usage distribution of a place tag should be the same in a cluster for all the scales. The method of cluster the tags

are proximity-based. For most of the scales the cluster is all the points within a certain region of neighborhood could be merged before points are parted. The calculation of the distance occurs after the points are merged. We define a threshold for merging method. The tags can be measure differently in different types of threshold. There are three types of variation in measurements. Measurements on accumulation of scale, in this method the measurement is calculated by sum of a list of sample of scales chosen from exponential sample method. The tag is place tag if the summed value is lower than the threshold. The value is entropy, thus we select the value which is lower than threshold. The lower entropy is more concentrated distribution for the tags. The strength of this measurement is to weight the scales. We can include multiple scales for the tags. Measurements on occurrence of scale, the stable values from a set of calculation based on multiple scales can be calculated. The qualifier stable is a standard to choose the scale from the set of all scales. If the value is l0% more than any points of scale, the tag can be classified as place tag. This method is based on the attribute of the cluster. That is the occurrence of the core set of tags tends to be a single cluster, which is stable and distinct to the other clusters. The tags other than place tags almost have the multiple strong clusters. If we merge all of these clusters, the final threshold computed should be larger than the first method.

- Measurement on accumulation of scale. In this method the measurement is calculated by sum of a list of sample of scales chosen from exponential sample method. The tag is place tag if the summed value is lower than the threshold. The value is entropy, thus we select the value which is lower than threshold. The lower entropy is more concentrated distribution for the tags. The strength of this measurement is to weight the scales. We can include multiple scales for the tags.
- Measurement on occurrence of scale. We find the stable values from a set of calculation based on multiple scales. The qualifier stable is a standard to choose the scale from the set of all scales. If the value is l0% more than any points of scale, the tag can be classified as place tag. This method is based on the attribute of the cluster. That is the occurrence of the core set of tags tends to be a single cluster, which is stable and distinct to the other clusters. The tags other than place tags almost have the multiple strong clusters. If we merge all of these clusters, the final threshold computed should be larger than the first method.
  We also need the alternation in the significant test. Like the first method, we select the tags which are lower than a threshold. We can also include multiple scales for the tags.
- Measurement Mixed. We combine the previous two methods as this method of measurement. Entropy we calculated after the sampling of the scales is the value we need to tell if the tags are a place tag or not. The set of the scales should be ordered in this scenario. In this method, the entropy should be close to zero, if the spatial distribution of the tags is more like a strong single cluster. Hence, the tags we are looking for, which are a small set of the tags, should have a smaller value computed than the tags which are not about describing a place or location. Similarly, the tags which have computed value smaller than the threshold are considered to be the place tags. We select those tags to mark the picture to be a spatial region, which is also discovered by the measurement methods.

2.4 Databases

Database management lies in the domain of computer science and software engineering for the purpose of maintaining data for easy and quick retrieval and updating. The data is kept in another program in the database with all the images, tags, and image features plus other relevant information, and all the algorithms are implemented with the help of queries to the database. There are many databases for image recognitions. The database often store geo-information as well. The GPS coordinates because the key to search for image in database therefore. In geo-database, we have many forms of geographic information available to search. In those databases, there are some key databases that support geo-location very well. Point of Interest (POI) geographical database provide geographic knowledge to image recognitions. It helps to train the machine learning algorithm for specific image which is for POI. Usually the POI image associates with an official name, geo-location information and demographic information. Querying a POI database, the accuracy of tagging system can be improved as well. However, POI database is relative small. Many pictures are not taken in POI. The POI database may not contain the sub region in the POI. So POI database is very limited in image recognition. Aerial image database provides the overlook of a certain place. The usage of aerial image is popular in Google Earth and Bing Bird's Eye. The important of aerial image database is that it provides the local environment of a certain place. We can learn the neighborhood of the places we are about to learn. Plus, the street view of Google maps is another source to learn the details of a place. However, the aerial image often covers a large area, which is not suitable for a specific location. For the places that are on the ground, the aerial image provides little clue to recognize the picture. The aerial image is only for study the environment of a large area. Existing pictures on the web can also be taken as a database of images and can be used for image recognition. Recently the usage of tagging system and picture sharing service is booming on the web. According to the existing web resources, it is easy to match a new picture to an existing one. The tagging system helps

people to mark a meaningful location to the picture, so matching picture will help us to discover the potential meaning of the picture as well.

2.5 Metadata

Metadata is a file with every picture containing basic information like the date and time at which the picture was taken, or with which camera the picture was taken at what resolution etc. The metadata of the pictures is valuable data for tagging process. The metadata for geo-location includes the GPS location (latitude and longitude) and heading information. The metadata is not limited to the information stored in the header of the image file. Some online service such as Flickr provides the service for users to pick up the geo-location where the pictures are taken. We can treat the information from third parties as the single data source for the geo-location for the pictures. There are two common schemas for storing geo information within an image file. RDF translation of the EXIF standard according to W3C-Exif (2003) contains the heading information and camera related data. It is a standard format stores the latitude and longitude. Dublin Core as presented in DCMI (2006) serves as the same as Basic Geo Vocabulary. In Dublin Core, the location context is expressed in label coverage. The location context is plain text composed of spatial location. For the common use, the location context is the postal address drilled down to the town name.

2.6 Geo-tagging

Flickr, the most popular photo sharing online service, added the functionality to allow users to geo-tagging to their picture. Usually a picture was taken at a time and a location. Tagging a picture with its location has a great value to photo sharing, business and social communication. With geo-tagging enabled, the entire picture can be connected in a way that everyone can easily browse and search for the view of a special physical location. Users are also benefit from discover related service, news and other resource more easily. The business can find better business opportunity by providing service based on location-aware technology. It is a great improvement to daily life that people can know a location better without even explores the location in person. Flickr provides users a service to tagging their own picture. From this abundant data of picture, we can associate the place and locations via the tagging relationship. The places share the common characteristics often are tagged with the same or similar text. The text-based (or tag-based) search is easy for the data mining process. Currently the commercial search engine and web photo services rely on the text mining, which is to associate the tags with images for better indexing and retrieval. The semantic analysis used in tag library helps to discover the relations between tags that are not explicit.

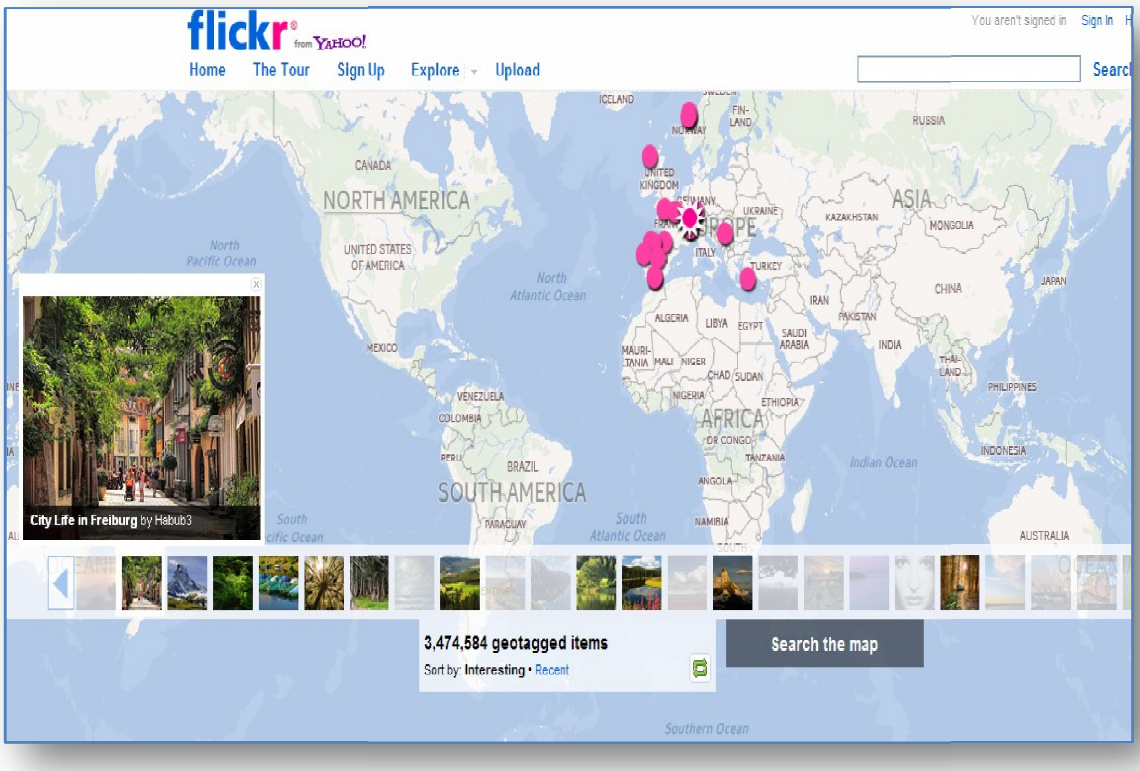


Figure 4.Flickr geo-tagging

2.7 Image Processing

There are two primary types of digital images one may be categorized as static images and the other as dynamic images. In static images the scene to be captured is not moving and the camera is also mounted at a fixed place, thus everything involved in static images is stationary. While dynamic images are those in which either multiple camera lenses are used to capture the image or any one or either the cameras or object is moving as stated by Alan (2011). A digital image is defined by 2 dimensional matrix containing pixels at x and y spatial coordinates and function of this x, y coordinates can be defined as the intensity or gray level of the image at any specific pixel position. To scan an artwork it requires dynamic image processing techniques to effectively design the scanner system. A scanner contains one or more than one camera lenses focusing at different viewpoints and cameras generally move with the help of digital motors to capture the complete object to be scanned. Digital image processing techniques are used to acquire and manipulate analog images such as paintings or drawings or other types of artwork according to Davies (2012). Some of the most commonly used image processing techniques are scaling of images, background subtraction, digital image subtraction, image normalization, profiling of images, filtering of images. All these concepts in the

digital image processing have their own purposes and can be utilized for different purposes.

2.7.1 Image filtering

Often the images acquired from camera or other imaging devices are not in the form that they can be used directly for the intended purposes. There may be many factors involved which can cause distortion in the process. These causes may be because of the variations in lightning or lamination while taking the pictures or different intensity levels in the picture. So, images during the process of capturing or during the analog to digital conversion of images processes should be used to make the images better and optimized for the intended use. Such process in digital image processing is known as the filtering process or image filtering as mentioned by Gonzalez and Wood (2002). There are different types of filters which are used extensively in the filtering process these are low pass filtering and high pass filtering. Low pass filtering only allows the low frequencies to pass through them and blocks the high frequency content of the image. These types of filters are commonly used in image averaging and to find the median of the images. These filters are used to reduce noise in the image and to enhance the quality of image removing extra pixel information from the image according to Maclean and Jernigan (1988). These filters can sometime blur the image depending on the fidelity of the original image.

2.7.2 High Pass Imaging Filters

High pass filters are the filters which allow only the high frequency content of the image to pass through them and it block the low frequency pixel value to pass through. The most important use of high pass filter includes edge detection or sharpening of the image. High pass filters can be modeled using first order derivatives or the second order differential equations. Filters are always implemented on an image by selecting a small part of the image also referred as a grid and then a matrix of same dimensions is designed contained the filter values. The filter matrix is then convoluted with the image grid or sub-spatial matrix of the image according to Schaar and Hanno (2007). Sobel and Laplace are the most commonly used high pass filter to detect edges from images.

Sobel Operator:

Sobel operator or Sobel filter is most commonly used in digital image processing especially in edge detection algorithms. It actually calculates the gradient of the image intensity or the gradient vector at each point of the image. The Sobel filter is usually implemented by convolving two small filters and separable filters with the image both in vertical and horizontal directions. Sobel operator consumes a lot of computation power as it moves both in the vertical and horizontal positions according to Kroon (2009). The Sobel operator consists of two different small sized kernels or matrix to be convolved with the image grid. As a result of the convolution both the horizontal and vertical approximations for the derivatives is calculated as mentioned by Hanno and Schaar (2007). Sobel operator can be calculated by defining a matrix value such as if we consider A in the following formula as the source image grid and Gx and Gy as the two sobel kernels then the convolution will take place as follows:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

As the two Sobel kernels are separable from one another and they can be decomposed as the products of a differential kernel and the averaging kernel. If decomposition is done then the Sobel operator calculates the gradient with smoothing of the image. The Gx can now be defined as follows according to Jahne and Korkel (1999):

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

The x coordinate defines the vertical movement of the filter during the convolution process, the Gy matrix can also be decomposed in the same manner. The decomposed matrix can enhance the computation speed thus making the Sobel operator more powerful and useful in the digital image processing techniques as thought by Gilbarg (2001). To calculate the exact magnitude of the gradient

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

Laplace Operator

Laplace is another differential operator mostly used I digital image processing for edge detection and image sharpening. It also calculates the gradient function at any given value of the pixel. The Laplacian of a function at any given pixel location is calculated by sum of partial differential equations according to Shubin (2001). The Laplace filter was first designed by a French scientist and mathematician Simon de Laplace in 1800's. He designed the operator initially to study celestial mechanics, where the Laplace operators when applied on the gravitational potential will result in mass density as mentioned by Lindeberg (1993). Later the Laplace operator was extensively used in digital image processing specifically in edge or blob detection algorithms. Most of the Laplace operator in digital image processing is based on the Laplacian of Gaussian formulas. The input image is convolved with by a Gaussian matrix or kernel which is based on the following formula.

$$g(x, y, t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/(2t)}$$

If we consider f(x,y) as the image function and g(x,y) as the Gaussian function then the convolution of both can result in Laplace filter implementation according to the research done by Bennett and Peng (2006).

$$L(x, y; t) = g(x, y, t) * f(x, y)$$

$$\nabla^2 L = L_{xx} + L_{yy}$$

The Laplacian when computed generally results in strong responses and dark objects up to the extent of

$\sqrt{2t}$ and strong negative responses are calculated for brighter objects. Laplace have a few problems as its result are highly dependent on sizes of objects and the relationship between them. Different multi scale approaches are utilized to make Lapalce operator independent of the relationships between different objects and their sizes in the image.

The gradient operation in the Laplace operator can also be used very effectively to detect sharp edges, changing of gray levels over space very rapidly can indicate the presence of an edge in an image. But when gray level are changed very slowly in the image usually when edges in the image are not sharp enough to be detected by Gaussian filter alone according to Gonzalez, Rafael and Richard (2006). So, it is better to use the Laplace operator instead of using the gradient operator directly on the image. The edges in an image can be obtained using a second order differential equation to find the zero crossings in edges present in an image.

For two dimensional image matrix the Laplace matrix or working kernel can be defined as

$$\mathbf{D}^2_{xy} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

This kernel is used when only the horizontal and vertical neighbors of the current pixel are included in the gradient calculations. To include the diagonals as well in the Laplace operator calculation there are many different same sized kernels available, the most commonly used kernel in Laplace operator for diagonals is

$$\mathbf{D}^2_{xy} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Gradient

Gradient of an image is categorized as directional change of color or gray level intensity in an image. Gradient of an image contains very useful information about the image and its calculation can help in analyzing the image in great detail as mentioned by Barrio, Herrero and Sanz (2009). In digital image processing concepts the gradient of an image gives information about gradual change or gradual blend of color from low intensity pixels to high intensity pixels for example the gradual change from white color to black color in certain direction in an image. As gradient is the calculation of progression of color from one point to another so it is sometimes also referred as color progression of an image. The gradient of a two dimensional image at any given pixel location can be found

by using a filter which calculates the intensities both in vertical and horizontal locations. The gradient is given by the direction from that given point to the largest intensity pixel in its neighbors and magnitude of the vector gives the rate of change in intensities from one pixel to another. The intensities of pixels are unique to their discrete points and there is no continuous relationship necessary between the intensities at different pixel locations. Because of this nature of digital images simply taking derivative of the intensity function may not help in the calculation of the gradient of digital image. Sobel operate is commonly used to calculate the vector and the magnitude of the gradient of digital image. Although the Sobel operator is not very accurate but it can provide sufficient information about the direction and rate of change in the gradient of any given digital image. Sobel operator uses a 3x3 grid thus the gradient is calculated only in 3x3 sub spatial grids from an image. Gradient of an image plays an important role in many of the image processing concepts and it is more commonly known in the edge detection algorithms. Many different types of approximations and calculations of an image can be done using the information provided by gradient calculations.

3 RESEARCH METHODOLOGY

3.1 Introduction

This chapter includes explanation of different methods and techniques used to design the desktop application. The application design contains concepts from different fields of computer and software engineering. The main fields of interest for the application design are as follows

Image processing

The field of image processing lies in the domain of signals processing in which the input to a system is an image on which different algorithms are applied and then the received output can be an image, a feature, or a set of different features. In our case, the input to the image processing system is an image while the output is a set of features after the implementation of different algorithms.

Artificial intelligence

Artificial intelligence is basically a field or domain in computer science that deals and aims at the creation of intelligent machines that can perform certain functionality after consideration testing. In our application, the program is designed after training and modifying for a large number of inputs, and now the algorithm has been developed in an artificially intelligent manner to handle a large number of unsupervised testing samples.

Machine learning

Machine learning basically lies within the domain of artificial intelligence that deals with the training of a set of input data for the training of the program before actually implementing it for the testing phase. The algorithms related to machine learning in this particular research topic and implementation are k-nearest neighbor, neural networks, and matching text finding algorithms.

Database management

Database management lies in the domain of computer science and software engineering for the purpose of maintaining data for easy and quick retrieval and updating. The data is kept in another program in the database with all the images, tags, and image features plus other relevant information, and all the algorithms are implemented with the help of queries to the database.

3.2 Tools required

The application design will be implemented using Microsoft C++ .net framework to design the interface for the application. Microsoft access will be used as the primary database management system and SQL queries will be used for communication between the application and the database.

3.2.1 Microsoft Visual Studio

Microsoft Visual Studio is development tool or to be more specific it is an integrated development tool from Microsoft which can be used to develop many of the console or desktop applications with graphical user interfaces. Microsoft Visual Studio includes many programming languages like C#, C++ and VB.NET. .net framework is the primary framework used for all the languages supported in the Visual Studio.

The .NET Framework is an integrated Windows component for the development and application of the next generation applications and XML Internet services, which includes two main components – the common language runtime and the .NET Framework class library. The .NET Framework basically works on the common language runtime. Runtime managing codes at execution time offers some primary services such as thread and memory management along with the promoted robustness and security. The class library of .NET Framework is an object – oriented collection of reuse types, which support the development of various applications including classical command-line or graphical user interface applications, and also the latest applications based on ASP.NET such as web Forms and XML Web services.

3.2.2 Microsoft Access

Office Access or Microsoft Access is a well known database management system (DBMS) manufactured by Microsoft. Access is a member of the complete Microsoft Office Suite package. Relational databases can be created using Microsoft Access, although it stores data in its own format which is based on the Microsoft Jet Database Engine. Access can also integrate data with other Microsoft products like Excel or Word. Microsoft Access can communicate with .net developed applications using OLEDB data library provide in the .net framework.

4 APPLICATION DESIGN

4.1 Introduction

The application is designed in C++ using Microsoft Visual Studio 2010 (dot Net framework version 4). The database to store image URLs is designed in Microsoft Access. There is no need to install any SQL server but Microsoft Access later than the 2003 version must be installed on the computer in order to view and manipulate the database. There are two different forms designed in the application, one form is used to provide interface to search images from the database and the other form searches images directly from Flickr. There must be an active internet connection available to run the application, the database search form also uses internet to display the searched images using the Flickr URL saved in the database.

4.2 Flickr API

The first and foremost requirement of the application is to gather data to create a database. Collection of images was chosen to be done from Flickr website keeping in mind the popularity and number of images available on Flickr website. Flickr provides much information with the found images, as discussed in the literature review that Flickr encourages its users to add a little description and a set of tags with the images they upload to the website. This makes Flickr a complete source of all the information that can be related with an image. With all the popularity unfortunately Flickr does not provide any application or an application programming interface (API) for .net framework. An API contains functions and features that can be used by other development tools to integrate applications with Flickr website to get information about the images stored on the Flickr website or the Flickr repository. For machine learning of the application, the application requires at least 10000 or more pictures in the local database. Collecting information from images manually and feeding them to the database one by one is not feasible and it required a lot of time just to complete the database. Flickr API or the Flickr.Net library is used in the application for the purpose. Flickr.Net library is an open source or free licensee library available which can collect data from the Flickr website. Use of Flickr.Net library to communicate with Flickr website violates no copyrights as Flickr does not prohibit the use of such libraries to communicate with Flickr website through an application. The Flickr.Net library is completely written in C# .net framework environment and can be accessed from .net framework 2.0 or above. The following functions from the Flickr.Net are used in the application.

```
#using <FlickrNet.dll>
private: Flickr ^f;
```

private: PhotoSearchOptions^ o;
private: PhotoCollection^ phCl;
o->Extras = PhotoSearchExtras::OriginalUrl |
PhotoSearchExtras::OwnerName | PhotoSearchExtras::Description |
PhotoSearchExtras::Tags | PhotoSearchExtras::Geo;

The above code to interface with the Flickr website will return images based on tags entered by the user in application interface. The information which will returned by the Flickr website will contain URL of the image, the name of the user who uploaded the image, description associated with the image, latitude and longitude of the images and complete set of tags that are linked with the image.

4.3 Image processing library

Microsoft .net framework does not offer built-in functionalities to perform basic image processing techniques like image resizing or applying filter to the source image. An open source .net framework library AForge.net is used to perform image processing techniques in this project. AForge.net is a library available for .net framework with free licence thus no copyrights have been violated while using the library. The library defines many math functions that can directly be applied on the images. The library also comes with certain image filters both high pass or low pass filters and can directly be applied to the image. There are other techniques available and image processing can also be done manually using basic .net framework functions but that results in a slow process and AForge.net library is optimized specially for the image processing algorithms thus performs the calculation at much greater speed and reliability.

4.4 Database design

Database for the application is completely designed using Microsoft Access but the database is only used as the central repository of data for the application and no application interface is designed in Microsoft Access. All the manipulation, deletion and creation of data in the database are done through C++ managed application. The application uses SQL queries to communicate with database file stored locally on the computer. Database contains different tables which are used to store data by the application and some tables are used to perform calculations to reach the desired search outputs. The image information retrieved from the Flickr website is stored in the local database. The Flickr table in database contains an ID (database primary key), IMAGEURL, TITLE, DESCRIPTION, USERID, LATITUDE, LONGITUDE, KEYWORDS/TAGS, IMAGELOCATION (image URL to directly fetch images from the Flickr website) and TAG_COUNT (to store total number of tags associated with an image). There are more than 2500 images stored in the database for a single place like Eiffel tower, Pisa tower etc. A total of more than 10000 rows of this data are stored in the database. The database design is as follows:

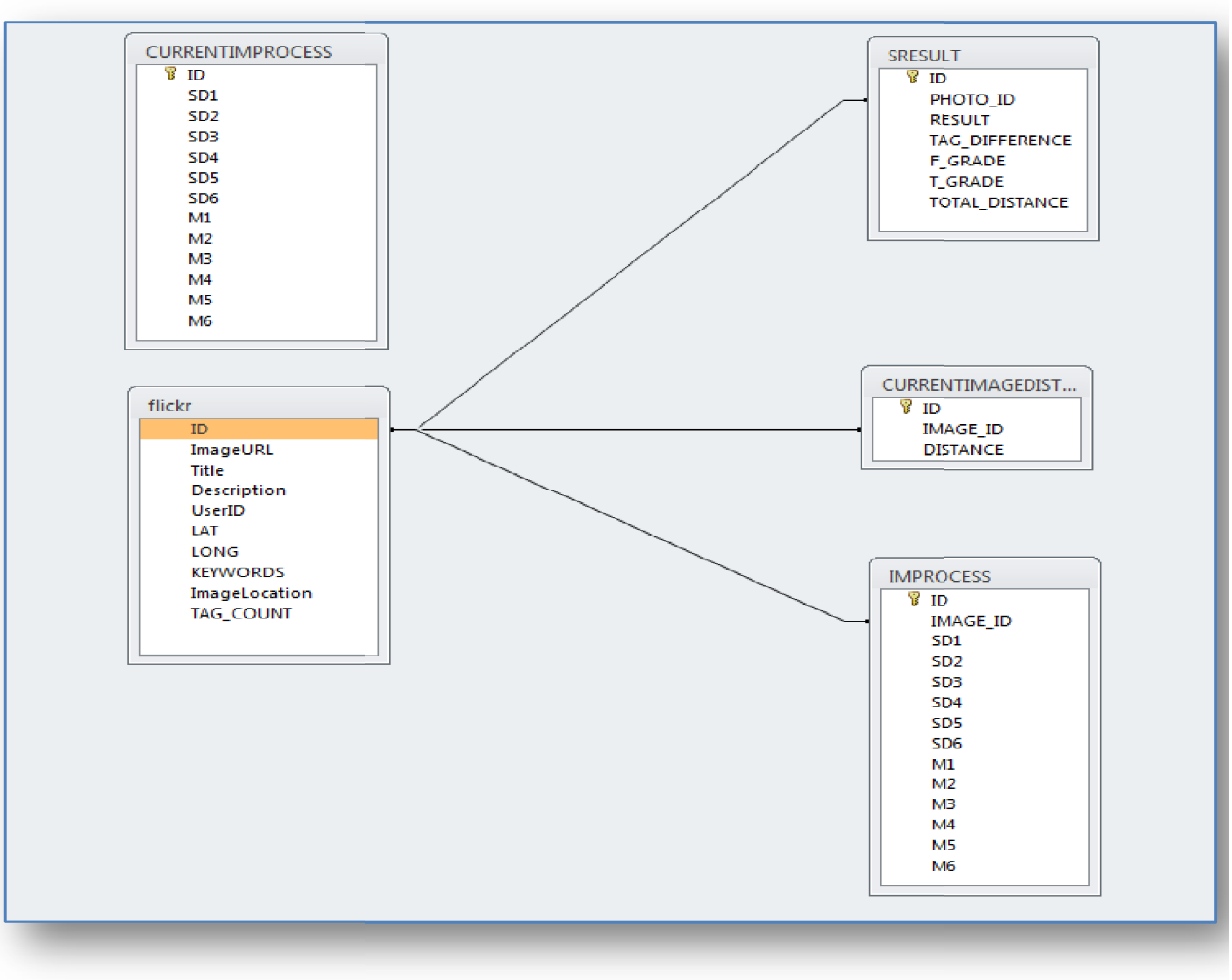


Figure 5. Database Design

Database contains five tables out of which four of the tables are connected with a central table while the other table is used for calculations done in the application. The Flickr table contains information about the image, it contains ID as the primary key for the table and thus ID is unique for all the records saved in the application. SRESULT table is used to save calculated distance of searched images using machine learning and artificial techniques in the application, all the results are saved in this table and final pictures are shown where the distance of images is closest to the searched image. All three other tables i.e. CURRENT IM PROCESS, IM PROCESS, CURRENT IMAGE DISTANCE are used for image recognition part of the application. Current IM PROCESS contains information about images already stored in the database in the Flickr table and this information contains results after image processing on the database images.

4.5 Tag search distance calculation

When a user enters a tag or a collection of tags, the application searches through all the images in the database and a distance is assigned to each of the found image. The distance calculation is based on the tag difference between the entered tags and the total number of tags associated with images in the database. The other important factor in the distance calculation is the number of found tags in the tag cloud of the images stored in the application. When a user enters a tag or a combination of tags to search in the database, the application searches the results in four steps as follows：

1. The application searches all the tags in the database and get results where all the tags are matched in the database e.g. The user entered Pisa, tower, Italy in the search text box, the application searches for images in the database where all three entered tags are found in the tag cloud of an image stored in the database. The value of 2 is given to such images, this value is chosen arbitrarily and shows that all the images found containing all the entered tags in its tag cloud.
2. The application then repeats the above step but this time searches for images containing any 2 tags from the entered combination in the image tag cloud. It only searches for images not found in the first step. The application assigns an arbitrary value of 4 for these images.
3. The application then repeats the above step but now searches for images where only 1 tag is found in the tag cloud. An arbitrary value of 6 is given to these images. Showing that these images have the least chances to be close to the desired search tags.
4. After this, the application calculates the total number of tags associated with the images found from the above three steps. Tag difference is calculated which is the total number of tags in the tag cloud of an image – the number of tags entered in the search text box. Maximum score chosen is 10 out of which up to 6 have already been assigned in the above three steps. The maximum tag difference found is then selected and stored in a variable, this is done to normalize the next calculations. The remaining score out of 4 is then calculated using the found tag differences as follows,

   (Current_tag_differences_from_found_image * 4)/(Maximum_tag_difference)

   This returns a value between 0 and 4. This value is then added to the score assigned in previous three steps. The resultant value is considered as the distance of the resultant image from the searched tags between 0 and 10. The distance values close to 0, or lower distance value means that the image is more close to the desired image on the basis of tags entered.

The application searches the entered tags in these steps, first step results the images where all the tags are found in the tag cloud of an image in the local database. This means that images containing all the entered tags in its tag cloud are shown as a result of this query. A minimum grade is given to these images which mean that it is most probable that the resultant images are nearest to the entered tags. ID of all the resultant images is stored in the database with a calculation on tag difference between the entered tags and the tags associated with the found image. The second part of calculation uses tag difference as the criteria to assign distance to the image e.g. if user entered three tags and all of the three tags are found in an image which contains 10 tags in its tag cloud. Another image is found which matches all the three tags out of 5 total tags in its tag cloud then the chances that the second image is more relevant to the desired output image are higher than the chances of first image being close to desired output. Once the calculation is done and all the information is saved in the application, second step of the search is performed. The application then searches the tags in pairs and checks the results already stored after the previous step. Only the results which are not already present in the database SRESULT table are stored in the database. The same method of distance calculation is applied to the results of this query and the results with assigned distance are stored in the database. Third step searches the least related images and images where only a single tag is found in the application database are displayed to the user with maximum distance from the searched images. A single found tag means that if a user enters three tags and any one of the tags is found in the images stored in the database, the application performs calculations on these resultant images but this step provides least relevant images with maximum distance from the desired image on the basis of entered tags. The following lines of code calculate and assign the distance to the resultant images:

```
for (int j=0; j <dataGridView1->Rows->Count-1; j++)
 {
        textBox2->Text += dataGridView1->Rows[j]->Cells[1]-
>Value->ToString();
        int ^td =  0;
        td = (int) (dataGridView1->Rows[j]->Cells[1]->Value) -
strarray->Length;
        OleDbCommand ^com = gcnew OleDbCommand("INSERT
INTO SRESULT (PHOTO_ID,RESULT,TAG_DIFFERENCE,F_GRADE)
VALUES ("+ dataGridView1->Rows[j]->Cells[0]->Value->ToString() +", 0,
"+ td +", 2) ");
    com->Connection =cn;


com->ExecuteNonQuery();

}
cn->Close();


if (strarray->Length>2)
{
```

```
                for (int i = 0; i < strarray->Length - 1; i++)
                {
                        for (int j = i + 1; j < strarray->Length; j++)
                        {
                                keyw = "[keywords] like '%" + strarray[i] +
"%' AND [keywords] like '%" + strarray[j] + "%'" ;

                                da = gcnew OleDbDataAdapter("SELECT
DISTINCT flickr.ID,flickr.TAG_COUNT FROM flickr WHERE NOT
EXISTS(SELECT PHOTO_ID FROM SRESULT WHERE flickr.ID =
SRESULT.PHOTO_ID) AND "+ keyw +"  ",cn);
                                table = gcnew DataTable();
                                da->Fill(table);

                                dataGridView1->DataSource= table;

                                cn->Open();

                                for (int j=0; j <dataGridView1->Rows-
>Count-1; j++)
                                {
                                        int  ^td =  0;
                                        td = (int) (dataGridView1->Rows[j]-
>Cells[1]->Value) - 2;
                                        OleDbCommand ^com = gcnew
OleDbCommand("INSERT INTO SRESULT
(PHOTO_ID,RESULT,TAG_DIFFERENCE,F_GRADE) VALUES ("+
dataGridView1->Rows[j]->Cells[0]->Value->ToString() +", 2, "+ td +", 4)
");
                                        com->Connection=cn;


                                        com->ExecuteNonQuery();

                                }
                                cn->Close();



                        }
                }
        }



        if (strarray->Length >1)
```

```
        {
                for (int i=0;i<strarray->Length;i++)
                {


                                        keyw = "[keywords] like '%" + strarray[i] +
"%'";

                                        da = gcnew OleDbDataAdapter("SELECT
DISTINCT flickr.ID,flickr.TAG_COUNT FROM flickr WHERE NOT
EXISTS(SELECT PHOTO_ID FROM SRESULT WHERE flickr.ID =
SRESULT.PHOTO_ID) AND "+ keyw +"  ",cn);
                                table = gcnew DataTable();
        da->Fill(table);

        dataGridView1->DataSource= table;

        cn->Open();

        for (int j=0; j <dataGridView1->Rows->Count-1; j++)
        {
                int  ^td =  0;
                td = (int) (dataGridView1->Rows[j]->Cells[1]->Value) - 1;
                OleDbCommand ^com = gcnew OleDbCommand("INSERT
INTO SRESULT (PHOTO_ID,RESULT,TAG_DIFFERENCE,F_GRADE)
VALUES ("+ dataGridView1->Rows[j]->Cells[0]->Value->ToString() +", 1,
"+ td +", 6) ");
                com->Connection =cn;


                com->ExecuteNonQuery();

        }
        cn->Close();
                }
        }


       da = gcnew OleDbDataAdapter("SELECT ID, TAG_DIFFERENCE,
F_GRADE, T_GRADE FROM SRESULT ORDER BY TAG_DIFFERENCE
DESC",cn);
        DataSet ^ds= gcnew DataSet();
        da->Fill(ds);

        dataGridView1->DataSource = ds->Tables[0];
```

```
        cn->Open();

        for (int j=0; j <dataGridView1->Rows->Count-1; j++)
        {
            int maxi= (int) (dataGridView1->Rows[0]->Cells[1]-
>Value);
            double  tg =  0;
            tg = ( (double) (int) dataGridView1->Rows[j]->Cells[1]-
>Value) * (double) (4) / (double)(maxi);

            double total_d =0;
            total_d = (int) dataGridView1->Rows[j]->Cells->Value +
tg;

            OleDbCommand ^com = gcnew
OleDbCommand("UPDATE SRESULT SET T_GRADE = "+ tg +",
TOTAL_DISTANCE = "+ total_d +" where id = "+ dataGridView1-
>Rows[j]->Cells[0]->Value->ToString() +"  ");
            com->Connection =cn;


            com->ExecuteNonQuery();

        }
        cn->Close();
```

4.6 Image Based Search

The application not only allows the user to search images from the database on the basis of tags only but a user can input an image and related images are found in the database using basic image processing techniques. AForge.Net library is used to implement basic image processing techniques in the application. When image information is fetched from the Flickr website and stored in the application some image properties are also saved in the application database in the IMPROCESS table. Text based image retrieval is most commonly used search method to search related images for a database. Text based images means that when a user enters an image to be searched the algorithm tries to automatically annotate the input image and find related images on the basis of tags. Automatic annotation of images is not yet designed to be very effective and such algorithms always work under human supervision in a semi-automatic mechanism i.e. user selects the correct annotations for the image or sometimes user has to provide some tags along with the image to be searched. The text based search is a fast mechanism but most of the image database around the world stores manual keywords which are not much reliable and annotating all the images available itself is time consuming and feasibility expensive task to perform. There are other methods that can also be utilized for the purpose like pattern matching by digital image processing techniques but that too is computationally expensive and it is nearly impossible to search a large database of images each time when a user enters an image to be searched. This application performs digital image processing on images at the time of entering images in the database and stores the calculation results about the image in a separate table linking with the main repository table by a unique primary key. Standard deviation and mean of an image can provide very useful information about the image like contrast of the image and information about distribution of the all the basic colors i.e. red, green and blue in an image. Standard deviation and Mean of R, G and B in an image can be an important link to predict the similarities in two different images. Also no runtime calculations are required like in the process of correlation where the input image is correlated with the source image every time we need to find the correlation between two images. An image when fetched from Flickr website for the local database is divided into six equal small rectangles and then mean and standard deviation of these rectangles is saved in the application. The purpose of segmenting the image in six equal parts is to minimize the calculation done on each image segment and it also enhances the efficiency of results found as now each segment of an image is compared instead of the complete image which can lead to wrong results. This process is done on all the images in the local database and the standard deviation and mean values of the entire six segments are stored in the database for future calculations thus saving calculation time at runtime when user

actually inputs an image to search related images from the database. The following lines of code perform this function in the application

```
ResizeNearestNeighbor ^ filter = gcnew ResizeNearestNeighbor(210, 140);
                bmp = filter->Apply(bmp);

                Rectangle rect;
                Bitmap ^ bmp1;

                String ^ sd="";
                String ^ m="";
                for (int i=0;i<2;i++)
                {
                        for (int j=0;j<3;j++)
                        {
                                rect= *gcnew Rectangle(j*70, i*70, 70, 70);

                                bmp1 = bmp-
>Clone(rect ,PixelFormat::Undefined);

                                ImageStatistics ^ stat = gcnew
ImageStatistics(bmp1);

                                Histogram ^ red = stat->Red;
                                Histogram ^ green = stat->Green;
                                Histogram ^ blue = stat->Blue;

                                double avgd= (red->StdDev + green-
>StdDev + blue->StdDev)/3;
                                double avgm= (red->Mean + green->Mean
+ blue->Mean)/3;

                                sd += avgd + ",";

                                m+= avgm +",";
```

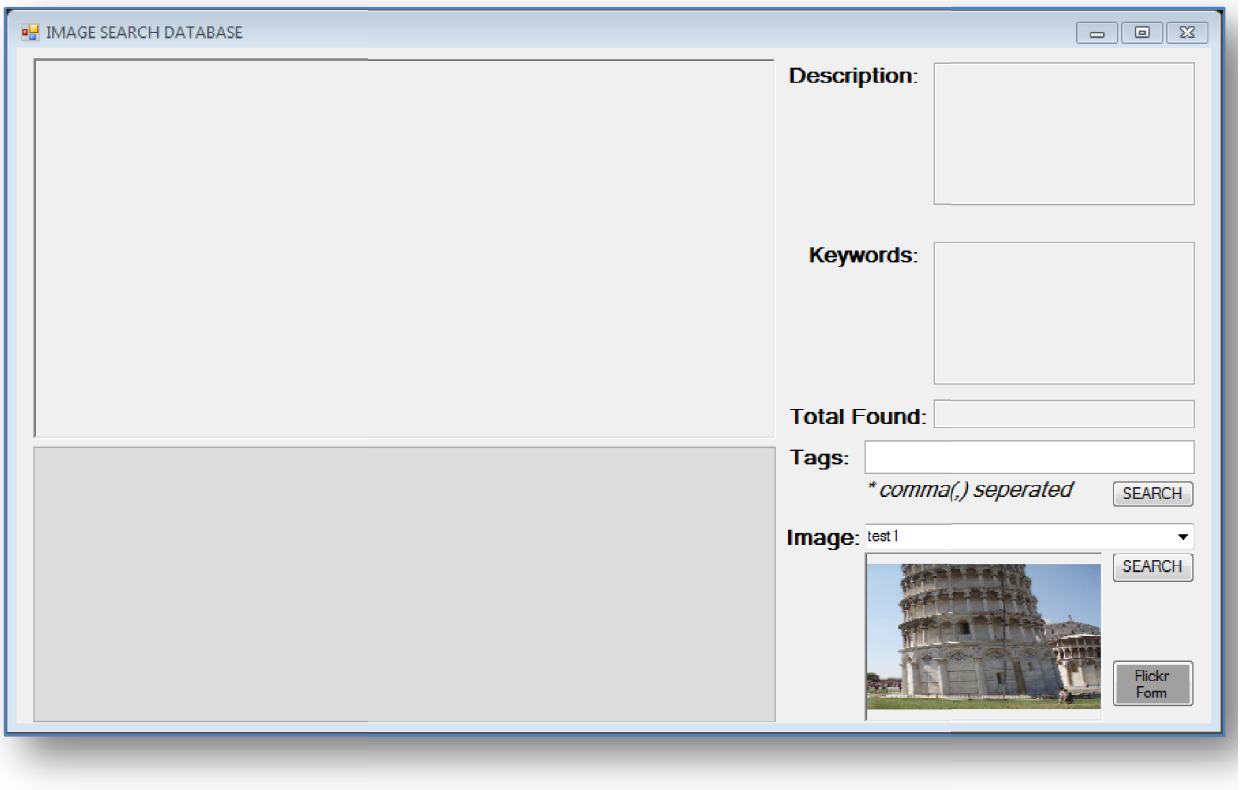
```
                }
            }
```

When a user enters an image to search in the database, the same process is done on the image and nearest images with those properties is displayed to the user. The input image is divided into six segments and standard deviation and mean of R, G and B is calculated for all the six segments. The resultants values are then compared with the values stored in the database and nearest values are found using the K nearest (KNN) algorithm and Euclidean distance. The closest results are displayed to the user as outputs.

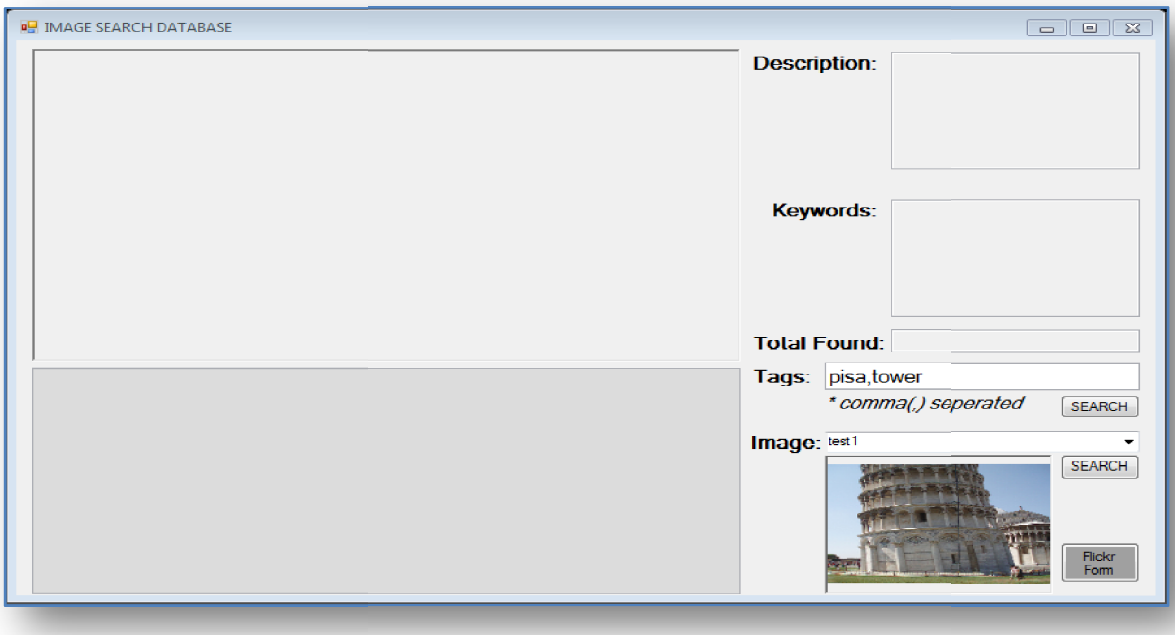
```
da = gcnew OleDbDataAdapter("SELECT * FROM
CURRENTIMPROCESS",cn);
                table = gcnew DataTable();
                da->Fill(table);
                dataGridView2->DataSource= table;

            double ed=0.0;

            for (int j=0; j <dataGridView1->Rows->Count-1; j++)
            {
                ed = 0.0;

                for (int k=2;k<14;k++)
                {

                    ed +=
(System::Math::Pow((double)dataGridView2->Rows[0]->Cells[k-1]->Value
- (double)dataGridView1->Rows[j]->Cells[k]->Value, 2));
                }
                ed = System::Math::Sqrt(ed);

                 com = gcnew OleDbCommand("Insert INTO
CURRENTIMAGEDISTANCE (IMAGE_ID,DISTANCE) VALUES
("+dataGridView1->Rows[j]->Cells[1]->Value->ToString()+","+ed+") ");
                com->Connection = cn;

                com->ExecuteNonQuery();

            }
```

5 RESULTS

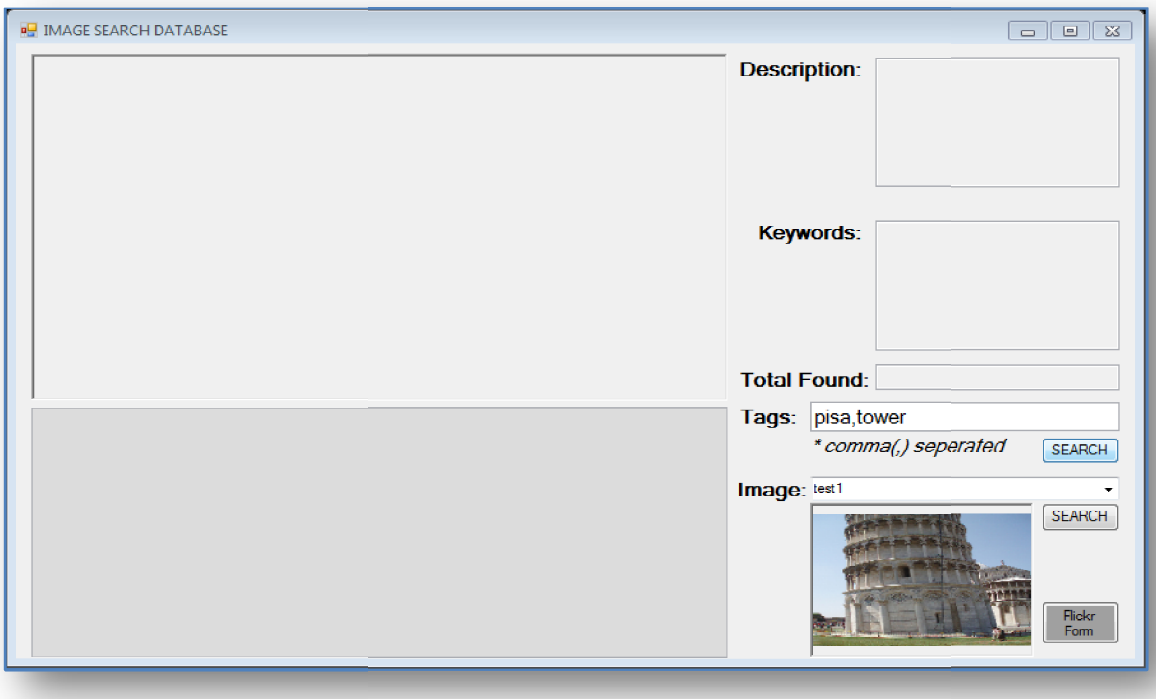
The designed application requires Microsoft Visual Studio 2010 with C++ to be installed on the computer to view and edit application. Access database ‘db1.mdb’ is located in the debug folder of the application. There is no need to change reference paths in the properties of the program from Visual Studio. All references are dynamically updated and ready to run at any computer. Following are the screenshots with brief details of each step to operate the application.



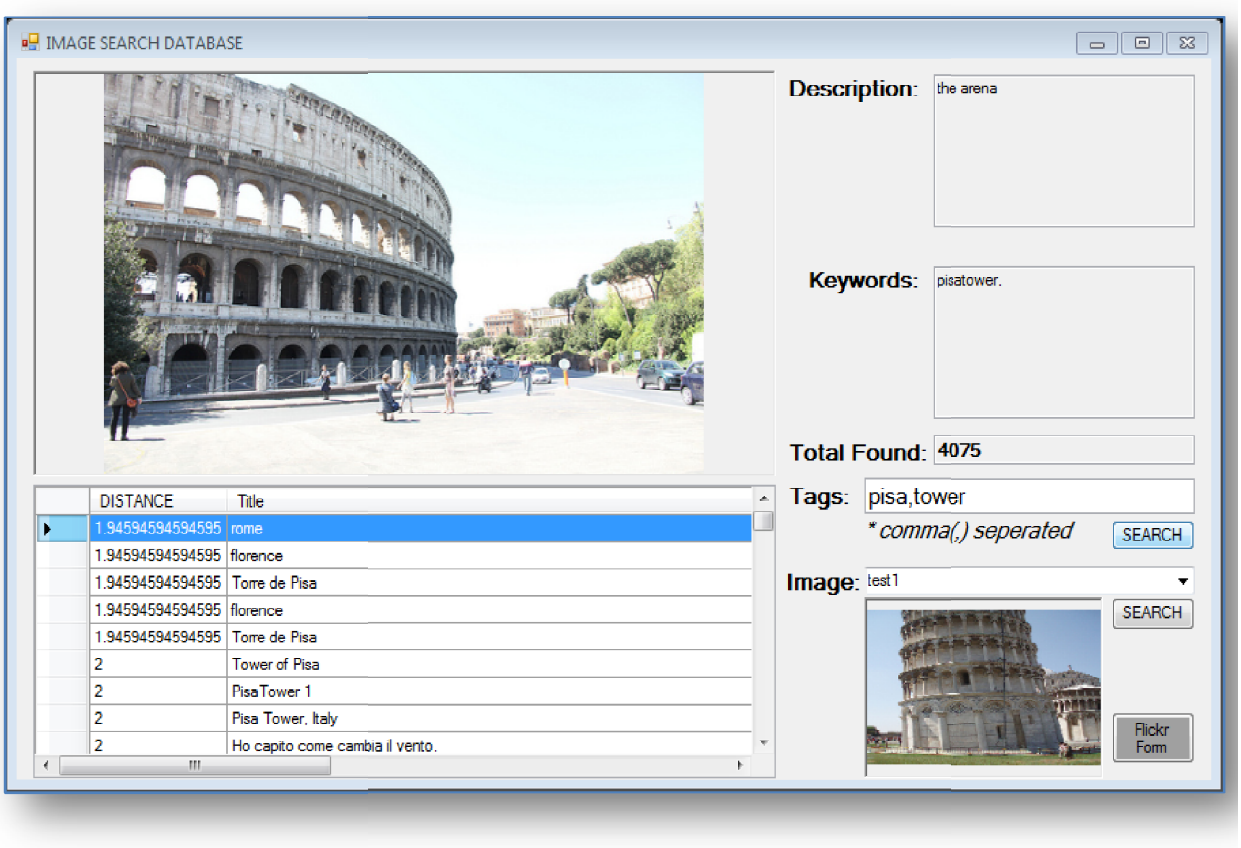
This is the main interface of the application and user can only enter data in the Tags field to search related results from the database. This form only search images from the image URLs stored in the database.



Multiple tags should be separated with a comma (,). Tags separated with a blank space are consider as a single tag and thus search the image on the basis of a single tag. The application automatically separates the tags with a comma between them so user only has to specify tags with a comma between multiple tags.



Once all the desired tags are entered, search button is pressed to look for related images in the database.



Once the search is complete, a list is displaced with image details. These details are the distance from searched tags, title of the searched image and Image Location i.e. URL of the image on Flickr network. The selected image is displayed in a picture box above the list. Other details of the selected image include description of the image, all keywords of the image this also includes keywords which are not matched with the searched tags. The interface also displays total number of images found from the database.

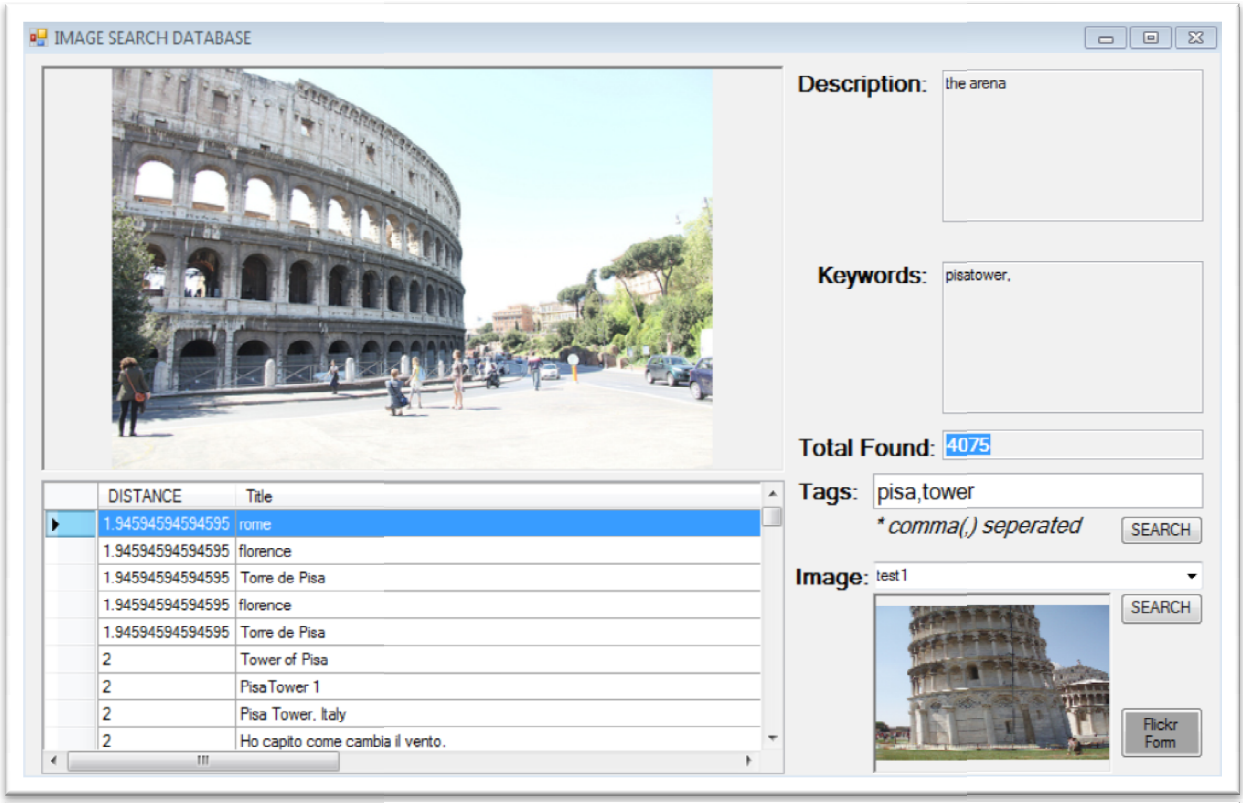


IMAGE SEARCH DATABASE
Description:
the arena
Keywords:
pisatower,
Total Found:
4075
Tags:
pisa,tower
* comma(,) seperated
SEARCH
Image:
test1
SEARCH
Flickr Form
DISTANCE
Title
1.94594594594595 rome
1.94594594594595 florence
1.94594594594595 Torre de Pisa
1.94594594594595 florence
1.94594594594595 Torre de Pisa
2 Tower of Pisa
2 PisaTower 1
2 Pisa Tower, Italy
2 Ho capito come cambia il vento.

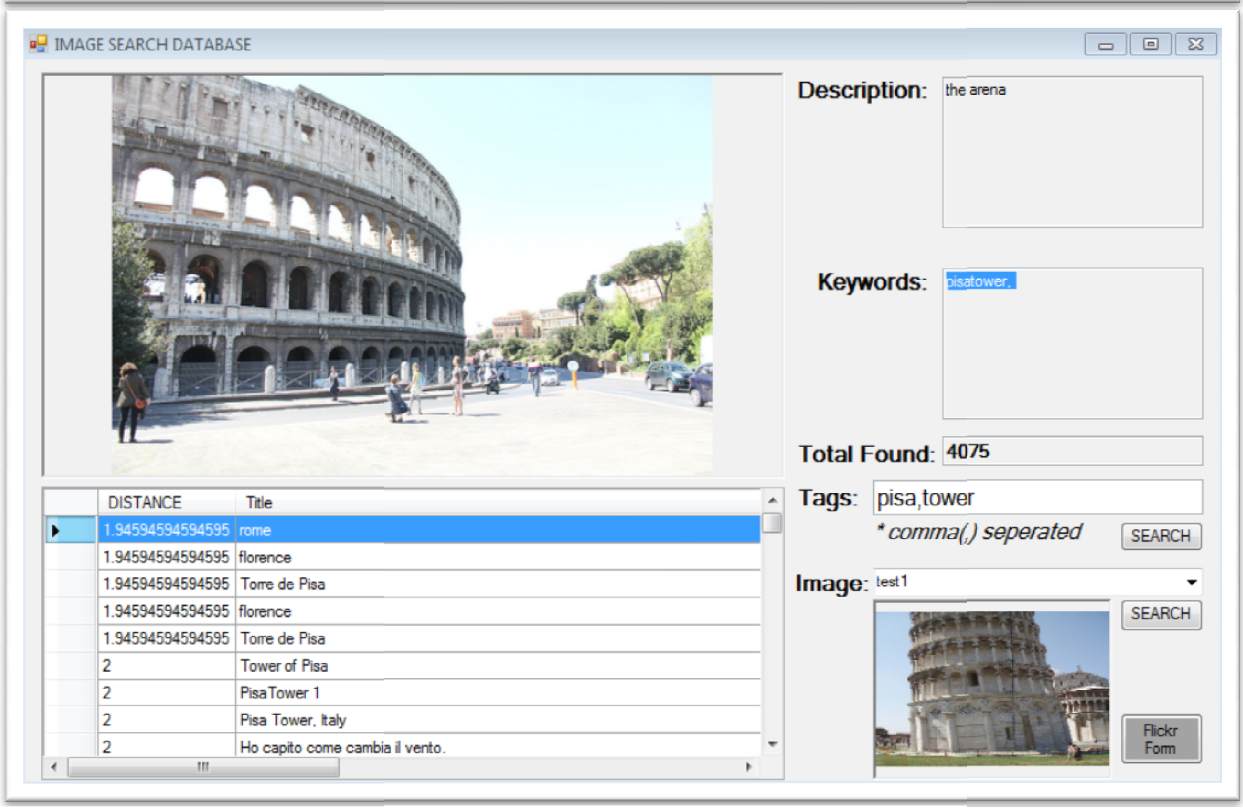


IMAGE SEARCH DATABASE
Description:
the arena
Keywords:
pisatower,
Total Found:
4075
Tags:
pisa,tower
* comma(,) seperated
SEARCH
Image:
test1
SEARCH
Flickr Form
DISTANCE
Title
1.94594594594595 rome
1.94594594594595 florence
1.94594594594595 Torre de Pisa
1.94594594594595 florence
1.94594594594595 Torre de Pisa
2 Tower of Pisa
2 PisaTower 1
2 Pisa Tower, Italy
2 Ho capito come cambia il vento.

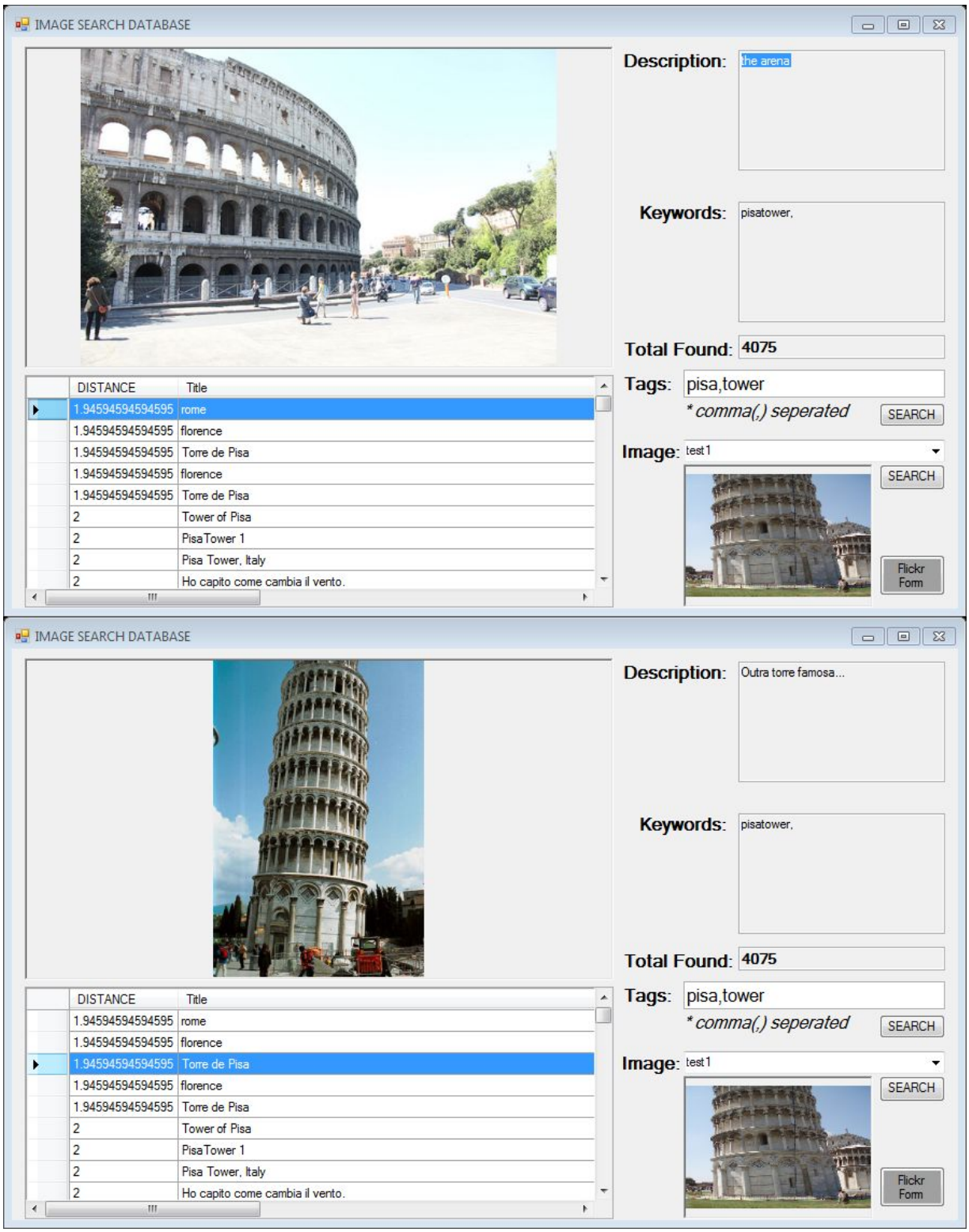
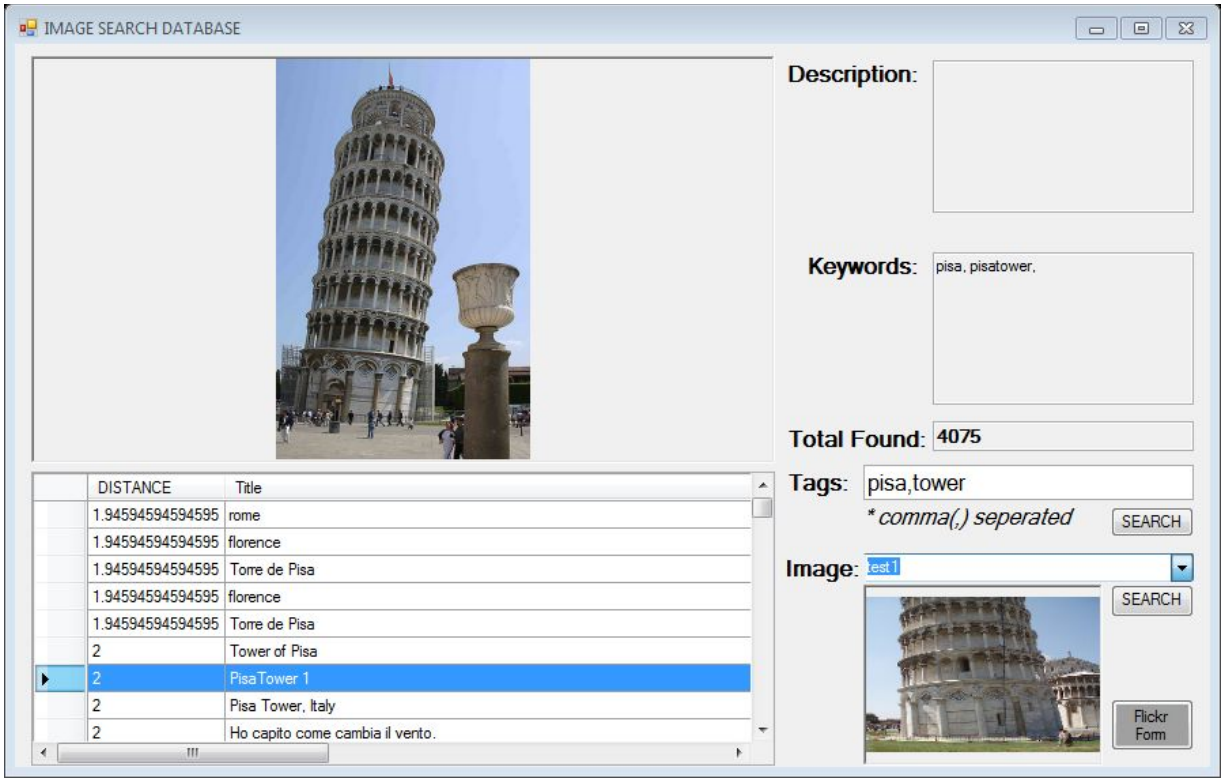


IMAGE SEARCH DATABASE
Description:
the arena
Keywords:
pisatower,
Total Found:
4075
Tags:
pisa,tower
* comma(,) seperated
SEARCH
Image:
test1
SEARCH
Flickr Form
DISTANCE
Title
1.94594594594595 rome
1.94594594594595 florence
1.94594594594595 Torre de Pisa
1.94594594594595 florence
1.94594594594595 Torre de Pisa
2 Tower of Pisa
2 PisaTower 1
2 Pisa Tower, Italy
2 Ho capito come cambia il vento.
IMAGE SEARCH DATABASE
Description:
Outra torre famosa...
Keywords:
pisatower,
Total Found:
4075
Tags:
pisa,tower
* comma(,) seperated
SEARCH
Image:
test1
SEARCH
Flickr Form
DISTANCE
Title
1.94594594594595 rome
1.94594594594595 florence
1.94594594594595 Torre de Pisa
1.94594594594595 florence
1.94594594594595 Torre de Pisa
2 Tower of Pisa
2 PisaTower 1
2 Pisa Tower, Italy
2 Ho capito come cambia il vento.

The above search is based on the tags. Only tags are matched to get desired output.

Images can also be searched by selecting desired image which is to be searched in the database. This search uses basic image processing techniques to get the results.



Below the tags filed, there is a combo box which includes 5 different images and user can select only one image at a time. The selected image is displayed in a picture box blow the combo box. When search is clicked the application starts matching the image with the images stored in the database. The output is populated with only the top 5 results found and in which the distance from image is less than 80 points.

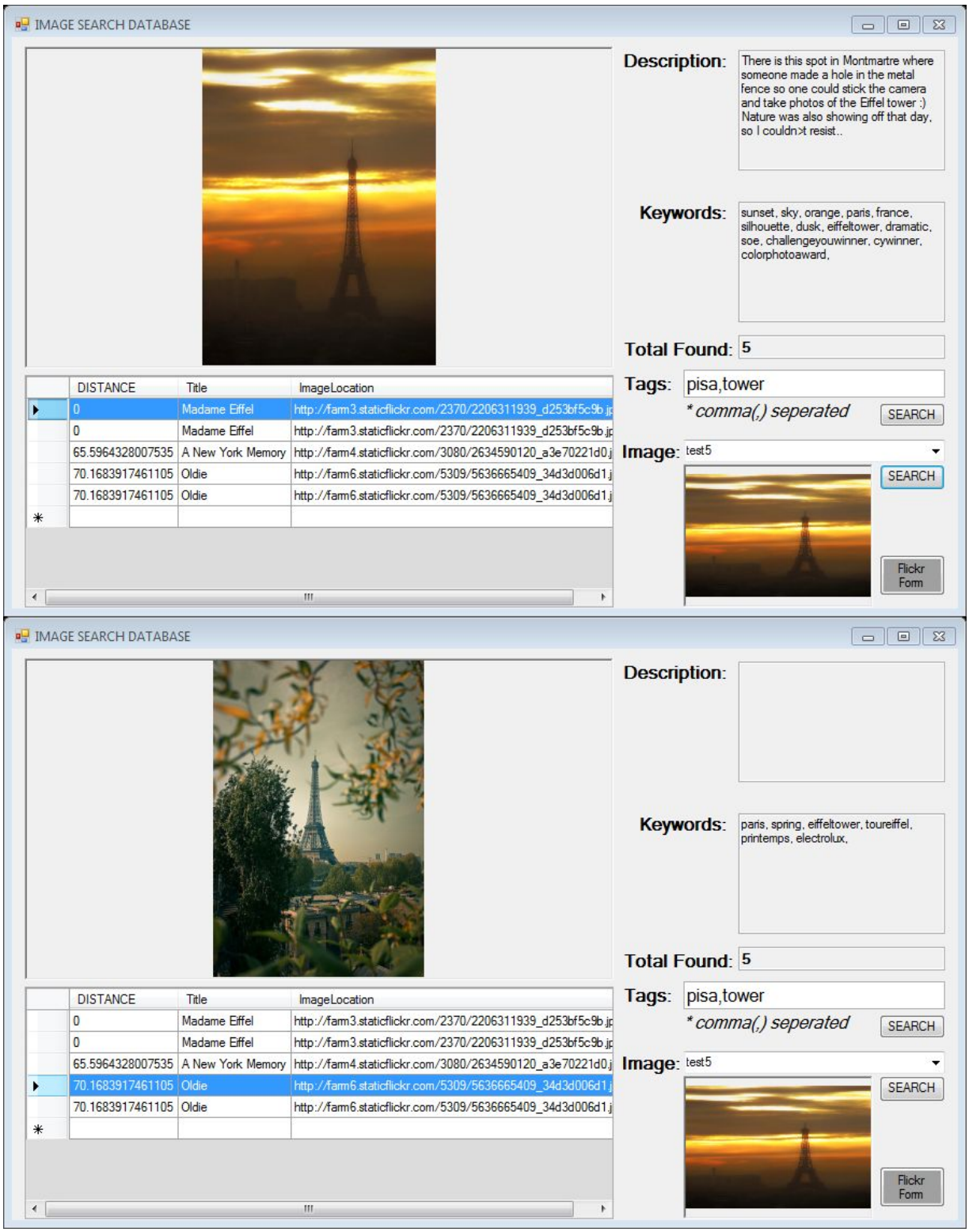
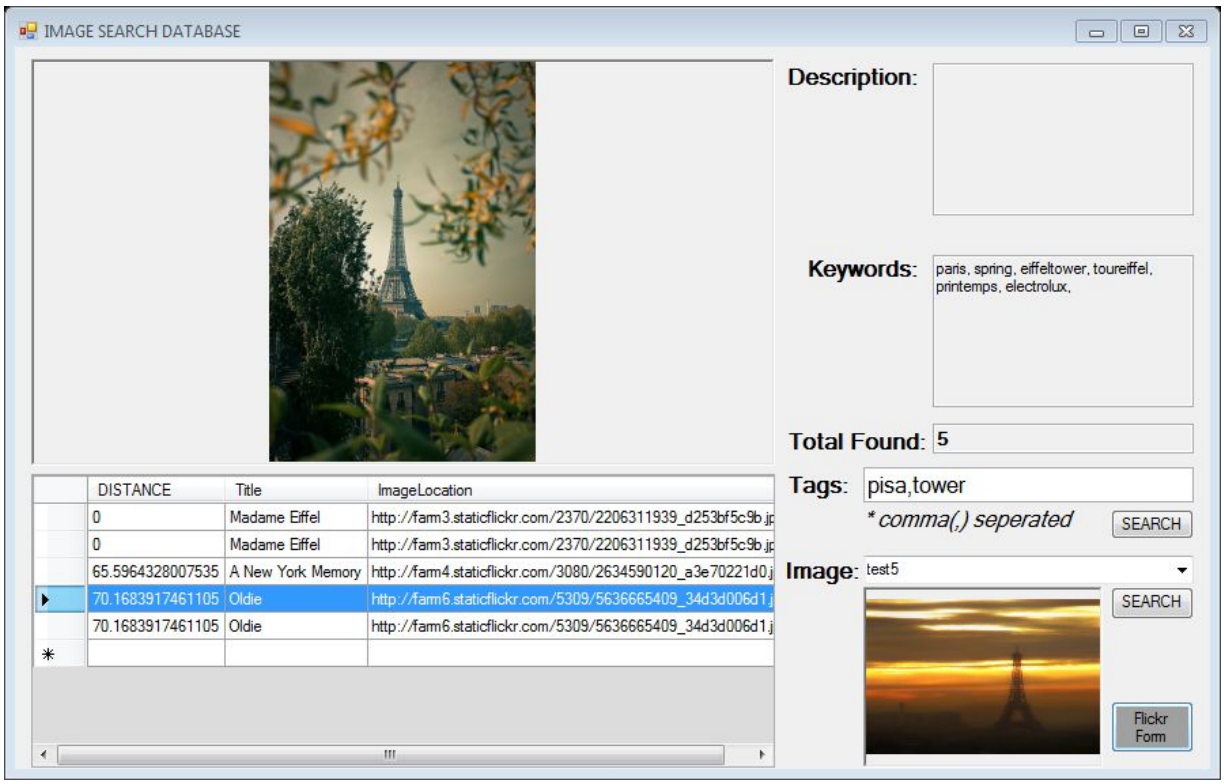
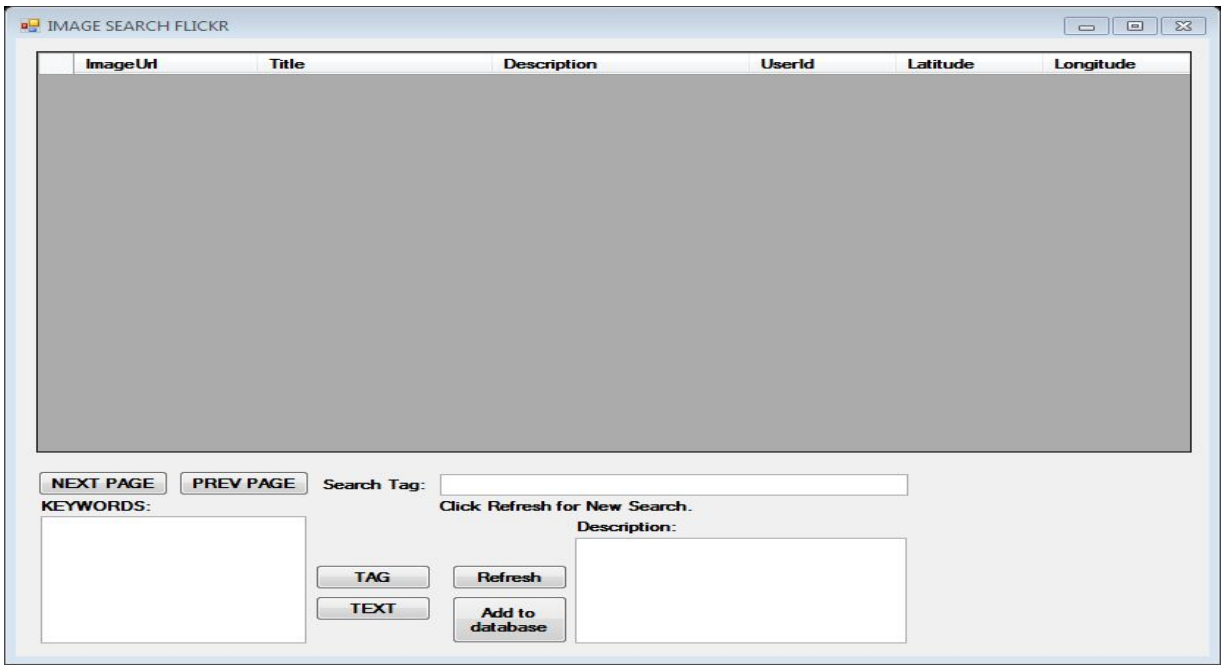


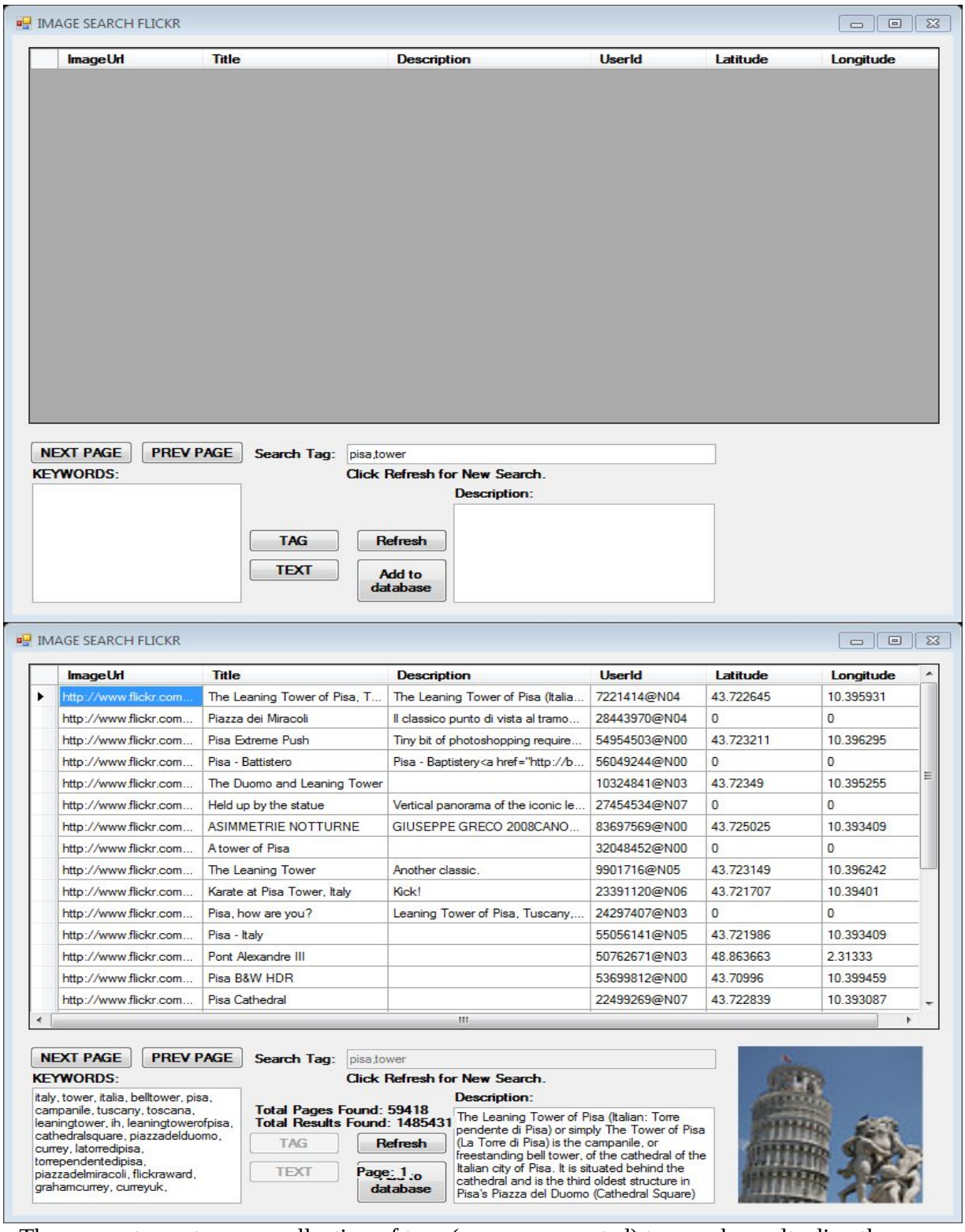
IMAGE SEARCH DATABASE
Description:
There is this spot in Montmartre where someone made a hole in the metal fence so one could stick the camera and take photos of the Eiffel tower :) Nature was also showing off that day, so I couldn>t resist..
Keywords:
sunset, sky, orange, paris, france, silhouette, dusk, eiffeltower, dramatic, soe, challengeyouwinner, cywinner, colorphotoaward,
Total Found:
5
Tags:
pisa,tower
* comma(,) seperated
SEARCH
Image:
test5
SEARCH
Flickr Form
DISTANCE
Title
ImageLocation
0
Madame Eiffel
http://farm3.staticflickr.com/2370/2206311939_d253bf5c9b.jp
0
Madame Eiffel
http://farm3.staticflickr.com/2370/2206311939_d253bf5c9b.jp
65.5964328007535
A New York Memory
http://farm4.staticflickr.com/3080/2634590120_a3e70221d0.j
70.1683917461105
Oldie
http://farm6.staticflickr.com/5309/5636665409_34d3d006d1.j
70.1683917461105
Oldie
http://farm6.staticflickr.com/5309/5636665409_34d3d006d1.j
IMAGE SEARCH DATABASE
Description:
Keywords:
paris, spring, eiffeltower, toureiffel, printemps, electrolux,
Total Found:
5
Tags:
pisa,tower
* comma(,) seperated
SEARCH
Image:
test5
SEARCH
Flickr Form
DISTANCE
Title
ImageLocation
0
Madame Eiffel
http://farm3.staticflickr.com/2370/2206311939_d253bf5c9b.jp
0
Madame Eiffel
http://farm3.staticflickr.com/2370/2206311939_d253bf5c9b.jp
65.5964328007535
A New York Memory
http://farm4.staticflickr.com/3080/2634590120_a3e70221d0.j
70.1683917461105
Oldie
http://farm6.staticflickr.com/5309/5636665409_34d3d006d1.j
70.1683917461105
Oldie
http://farm6.staticflickr.com/5309/5636665409_34d3d006d1.j



FLICKR form:

Flickr form can be accessed using the ‘Flickr form’ button on main application interface.





The user enters a tag or a collection of tags (comma separated) to search results directly from the Flickr website. User can view all the pages returned from Flickr website with information about the title of image, user id of the user who uploaded the image, a brief

description of image and the complete set of tags associated with the image. If the user presses the ‘Add to database’ button on Flickr form, the application automatically adds the top 2500 results returned by the Flickr website these results are not shown to the user and directly imported to the database. A total of 10000 images are currently imported in the local database. The images stored in the database are form different locations around the world which includes Pisa tower Italy, Eifel tower, Statue of liberty and Empire State Building.

Limitations of the project are images stored in the database are very limited and about a very few different locations. Future work may include the learning of relationships between the tags and images to define sound algorithm for automatic annotation of images which will greatly help in image recognition processes and also improve the accuracy of results found related to images given by the user.

6 EVALUATION

The project was aimed at the formation of a system for the detection of images corresponding to the input tags or image data from the online resource for images and labels named **Flickr.** The previous methods for solving these problems considered only the image data and utilized the image processing algorithms for the computation of similar images in the online databases. This project was also intended to focus on the information stored in the metadata in the forms of tags that can help in detection of relevant similar images from the online image databases.

The method developed for the implementation of the project contained algorithms from machine learning, image processing, artificial intelligence, and database management plus access. The algorithms were developed with the intention of computing similar data from the online resources with improved efficiency and accuracy. Almost all of the previous methods utilized graph-based search and walk-through to move through the related images but did not use image processing techniques. All of those researches were conducted on the data available from research institutes with the relationships developed between images already with probabilities defined by them, but the real-time applications are entirely different than those implementations. This project was developed by keeping in mind the real world applications, therefore no data has been extracted from research institutes rather complete information has been extracted from Flickr which makes this application salable in the market as well.

The aim of the project was to implement image processing algorithms for the detection of similar images if the input data is an image, and to implementation machine learning and pattern recognition algorithms for the detection of similar images if the input data is in the form of tags. Both of the parts are implemented in the project with 100% accuracy as depicted by the results section that the correct images are found out every time. The project is developed for around 5 cities with around 10000 images but many further improvements will be required in algorithm design for the implementation for all the places around the world. The aims and objectives of the project have been achieved, and the implementation has been completely done with the help of Visual Studio, Aforge library, and Flickr API.

6.1 Limitations

There are some limitations in the product development in terms of the implementation of the project. The images stored in the database are around ten thousand images and they are of around 5 different places. The implementation of the project on a large scale needs

a much bigger database of images which is hard to implement. The project is developed in Microsoft Access which limits the number of entries to eight thousand. So, there is a need to switch the application to other platforms for database management. The speed of the calculations needs to be increased when the database size would increase. The project is limited to do calculations completely via processor. The implementation of the image processing algorithms can be done on the graphical processing unit with the help of CUDA language to divide the processing between the processor and GPU for the sake of quickening up the speed and efficiency which will help to get over this limitation.

Several limitations have been faced during the implementation of the project. The data has been extracted from the online website for photo sharing called as Flickr. There is no official application programming interface provided by the website for the extraction of data to the system via software. The only available application programming interfaces are made by the third-party without any documentation, and they are all made for the window based application. There is no API for .Net which was a limitation in the start. The API had to be converted completely to adjust to .Net framework for the extraction of data.

Another limitation of the project is that the tags are uploaded with the image by normal people. So it is not necessary that the tags are relevant to the image, and sometimes they are completely irrelevant which causes limitations in the project. Algorithms need to be developed for extracting only the relevant tags and discarding the irrelevant tags. The project has also overcome this problem to a certain extent by giving probabilities to the tags but better and efficient algorithms still need to be developed.

The image processing algorithms need to be fast because the time consumed in the searching of a huge database, and for comparing the image to all the images stored in the database is huge. Therefore, the image features of the images stored in the databases have been saved in the form of integers, but not all the features can be saved in this manner which causes a huge limitation in the project.

6.2 Further research

The mechanism developed for this particular project does not involve a supervised learning and training phase before stepping into the testing phase. The training phase is important in the implementation of the project in the domain of machine learning. The future researches in the field should focus on the development of algorithms that have a training set of data which improves the performance of the implementation by iterative methods. A cycle of testing and training phases should be repeated unless the required accuracy on a large scale is achieved. Iterative techniques of image annotation with the help of bi-relational graphs and probabilistic techniques are under development by certain scholars as depicted in the latest research paper regarding the field of machine learning. The probabilistic methods with the inclusion of Bayesian and k-nearest neighbor algorithms of pattern recognition, and the graph traversal algorithms with linkages between similar type of image and data classes can largely help in the process of detecting similar data from the online resources. The further research witnessed in this field has been quite low in efficiency, therefore the implementation of these latest researches by turning the execution of image processing algorithms to the graphics processing units instead of processors can help largely in terms of efficiency. Therefore, these changes should be considered in the designing of techniques and algorithms in the future researches for the annotation of unlabeled images.

Image annotation for unlabeled images is an emerging topic for research. This research is normally focused on attaching tags to the unlabeled images. First of all, a graph is created for the image data set and then a graph is created for the class of labels. The labels in the graphs are associated with each other with certain probabilities and so are the images. After that, a bi-relational graph is created for the relations between the labels and the images. Whenever a new image data is arrived, the traversal through the graph is done from image data graph to the label graph on the basis of probabilistic modeling for the purpose of attaching labels with the images. This particular research time can give rise to a large number of important applications in the future; therefore a lot of research should be focused on the image annotation of the unlabeled images. The project developed in this report can provide the basis for conducting such researches to a large extent.

7 REFRENCES

1. Amazon Mechanical Turk. http://www.mturk.com/
2. Erick Schonfeld. Who Has The Most Photos Of Them All? Hint: It Is Not Facebook. TechCrunch, 2009.
3. Dong and B. Bhanu. Active concept learning for image retrieval in dynamic databases. In ICCV ’03: Proceedings of the Ninth IEEE International Conference on Computer Vision, page 90, 2003.
4. Ko and H. Byun. Probabilistic neural networks supporting multi-class relevance feedback in region-based image retrieval. In ICPR ’02: Proceedings of the 16th international
5. Conference on Pattern Recognition (ICPR’02) Volume 4, page 40138, 2002.
6. Y. Rui and T. S. Huang. Relevance feedback techniques in image retrieval. Springer-Verlag, London, UK, 2001.
7. S. Tong and E. Chang. Support vector machine active learning for image retrieval. In MULTIMEDIA ’01: Proceedings of the Ninth ACM International Conference on Multimedia, pages 107–118, 2001.
8. S. Zhou, Y. Rui, and T. Huang. Exploration of Visual Data. Kluwer Academic Publishers, 2003.
9. Trattner, C., Korner, C., Helic, D., 2007, “Enhancing the Navigability of Social Tagging Systems with Tag Taxonomies”
10. Davies E., 2012, “Computer and Machine Vision”, 4th Edition
11. Gonzalez and Woods, 2002, “Digital image processing”, 2nd edition, Prentice Hall
12. McLean, G., and Jernigan, E., 1988, "Hierarchical edge detection, Computer Vision, Graphics, and Image process”, vol. 44, pp. 350-366.
13. Scharr and Hanno, 2000,” Optimal Operators in Digital Image Processing”
14. Kroon, D., 2009, “Numerical Optimization of Kernel Based Image Derivatives”
15. Scharr and Hanno, 2007, “Optimal Filters for Extended Optical Flow”
16. Jahne,B., and Korkel, B., 1999, “Principles of filter design. In Handbook of Computer Vision and Applications”
17. T. Rattenbury, M. Naaman, Methods *for extracting place semantics from Flickr tags*, ACM TransWeb, Volume 3 Issue 1, January, ACM New York, NY, USA, Article No.1, 2009.

18. G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, F. Dellaert, *Detecting and matching repeated patterns for automatic geo-tagging in urban environments.* In Proceedings of IEEE CVPR, pp.1-7, 2008.
19. S. Lohmann, J. Ziegler, and L. Tetzla, *Comparison of tag cloud layouts: Task-related performance and visual exploration,* In Proceedings of the 12th IFIP conference on Human-Computer interaction(INTERACT), pp.392-404, 2009.
20. B. Epshtein, E. Ofek, Y. Wexler, and P. Zhang, *Hierarchical photo organization using geo-relevance,* in Proc. GIS, Article No.18, 2007.
21. J. Yuan, J. Luo , H. Kautz, Y. Wu ,*Mining GPS traces and visual words for event classification*. MIR '08 Proceedings of the 1st ACM international conference on Multimedia information retrieval, ACM New York, NY, USA ,pp. 2-9,2008.
22. C. Torniai, S. Battle, S. Cayzer, *Sharing, discovering and browsing geotagged pictures on the web,* HPL-2007-73,2006.
23. Y. Hassan-Montero, V. Herrero-Solana, *Improving Tag-Clouds as Visual Information Retrieval Interfaces*, In Proceedings of Multidisciplinary Information Sciences and Technologies (InSciT), Merida, Spain, 2006.
24. Marlow, C. et al, *Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead,* WWW2006, Edinburgh, UK, May 22–26, 2006.
25. S. Bateman, C. Gutwin, M. Nacenta, *Seeing Things in the Clouds: The Effect of Visual*
26. *Features on Tag Cloud Selections*, In: Proc. of the 19th ACM conference on Hypertext and
27. Hypermedia, ACM Press, New York, pp. 193–202, 2008.
28. J. Zhu, Z. Nie, J. Wen, B. Zhang, W. Ma, *Simultaneous record detection and attribute* labeling in web data extraction, In: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, New York , pp. 494-503,2006.
29. A.W. Rivadeneira, , D.M. Gruen, , M.J. Muller, D.R. Millen, *Getting our Head in the*
30. *Clouds: Toward Evaluation Studies of Tagclouds*. In: Proc. of the SIGCHI Conference on

31. Human Factors in Computing Systems, ACM Press, New York , pp. 995–998, 2007.

8 APPENDIX

Local Database Search: Code

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {

        OleDbConnection ^cn = gcnew
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=\db1.mdb;Persist Security Info=False");

        array<String^>^strarray = gcnew array<String^>(1);

        strarray= textBox1->Text->ToString()->Split(',');

        String ^keyw= gcnew String("a");

        OleDbCommand ^comd= gcnew OleDbCommand("DELETE *
FROM SRESULT");
        comd->Connection = cn;

        cn->Open();

        comd->ExecuteNonQuery();

        cn->Close();

        for (int i=0;i<strarray->Length;i++)
        {
            if (i==0)
            {
                keyw = "[keywords] like '%" + strarray[i] + "%'";
            }

            else
            {
                keyw += " AND [keywords] like '%" +strarray[i]
+"%'";
            }

        }
```

```
        OleDbDataAdapter ^ da = gcnew OleDbDataAdapter("SELECT
ID,TAG_COUNT FROM flickr where "+ keyw +" ",cn);

        DataTable^ table = gcnew DataTable();
        da->Fill(table);

        dataGridView1->DataSource= table;


        cn->Open();



       for (int j=0; j <dataGridView1->Rows->Count-1; j++)
        {
                textBox2->Text += dataGridView1->Rows[j]->Cells[1]-
>Value->ToString();
                int  ^ td =  0;
                td = (int) (dataGridView1->Rows[j]->Cells[1]->Value) -
strarray->Length;
                OleDbCommand ^ com = gcnew OleDbCommand("INSERT
INTO SRESULT (PHOTO_ID,RESULT,TAG_DIFFERENCE,F_GRADE)
VALUES ("+ dataGridView1->Rows[j]->Cells[0]->Value->ToString() +", 0,
"+ td +", 2) ");
           com->Connection =cn;


        com->ExecuteNonQuery();

        }
        cn->Close();


        if (strarray->Length>2)
        {
                for (int i = 0; i < strarray->Length - 1; i++)
                {
                        for (int j = i + 1; j < strarray->Length; j++)
                        {
                                keyw = "[keywords] like '%" + strarray[i] +
"%' AND [keywords] like '%" + strarray[j] + "%'" ;

                                da = gcnew OleDbDataAdapter("SELECT
DISTINCT flickr.ID,flickr.TAG_COUNT FROM flickr WHERE NOT
EXISTS(SELECT PHOTO_ID FROM SRESULT WHERE flickr.ID =
SRESULT.PHOTO_ID) AND "+ keyw +"  ",cn);
```

```
                        table = gcnew DataTable();
                        da->Fill(table);

                        dataGridView1->DataSource= table;

                        cn->Open();

                        for (int j=0; j <dataGridView1->Rows-
>Count-1; j++)
                        {
                                int ^td = 0;
                                td = (int) (dataGridView1->Rows[j]-
>Cells[1]->Value) - 2;
                                OleDbCommand ^com = gcnew
OleDbCommand("INSERT INTO SRESULT
(PHOTO_ID,RESULT,TAG_DIFFERENCE,F_GRADE) VALUES ("+
dataGridView1->Rows[j]->Cells[0]->Value->ToString() +", 2, "+ td +", 4)
");
                                com->Connection=cn;


                                com->ExecuteNonQuery();

                        }
                        cn->Close();



                }
        }
}



if (strarray->Length >1)
{
        for (int i=0;i<strarray->Length;i++)
        {


                        keyw = "[keywords] like '%" + strarray[i] +
"%'";

                        da = gcnew OleDbDataAdapter("SELECT
DISTINCT flickr.ID,flickr.TAG_COUNT FROM flickr WHERE NOT
```

```
EXISTS(SELECT PHOTO_ID FROM SRESULT WHERE flickr.ID =
SRESULT.PHOTO_ID) AND "+ keyw +"  ",cn);
                        table = gcnew DataTable();
        da->Fill(table);

        dataGridView1->DataSource= table;

        cn->Open();

        for (int j=0; j <dataGridView1->Rows->Count-1; j++)
        {
                int  ^td =  0;
                td = (int) (dataGridView1->Rows[j]->Cells[1]->Value) - 1;
                OleDbCommand ^com = gcnew OleDbCommand("INSERT
INTO SRESULT (PHOTO_ID,RESULT,TAG_DIFFERENCE,F_GRADE)
VALUES ("+ dataGridView1->Rows[j]->Cells[0]->Value->ToString() +", 1,
"+ td +", 6) ");
                com->Connection =cn;


                com->ExecuteNonQuery();

        }
        cn->Close();
                }
        }


       da = gcnew OleDbDataAdapter("SELECT ID, TAG_DIFFERENCE,
F_GRADE, T_GRADE FROM SRESULT ORDER BY TAG_DIFFERENCE
DESC",cn);
        DataSet ^ds= gcnew DataSet();
        da->Fill(ds);

        dataGridView1->DataSource = ds->Tables[0];

        cn->Open();

        for (int j=0; j <dataGridView1->Rows->Count-1; j++)
        {
                int maxi= (int) (dataGridView1->Rows[0]->Cells[1]-
>Value);
                double  tg =  0;
                tg = ( (double) (int) dataGridView1->Rows[j]->Cells[1]-
>Value) * (double) (4) / (double)(maxi);
```

```
                double total_d =0;
                total_d = (int) dataGridView1->Rows[j]->Cells->Value +
tg;

                OleDbCommand ^ com = gcnew
OleDbCommand("UPDATE SRESULT SET T_GRADE = "+ tg +",
TOTAL_DISTANCE = "+ total_d +" where id = "+ dataGridView1-
>Rows[j]->Cells[0]->Value->ToString() +"  ");
                com->Connection =cn;
```