Diversifying Relevant Search Results from Social Media Using Community Contributed Images

by

Vaibhav Kalakota

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2020 by the
Graduate Supervisory Committee:

Ajay Bansal, Chair
Srividya Bansal
Michael Findler

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

Availability of affordable image and video capturing devices as well as rapid development of social networking and content sharing websites has led to creation of new type of content, Social Media. Manual assessment of the relevance of these publicly available images to a particular query is not feasible due to the immense amount of data captured and shared daily on these social media platforms. As a result, the automated optimization of image retrieval results gains constantly in importance. Next to relevance, the aspect of diversification of retrieval results plays a crucial role in order to reduce the redundancy in the retrieved images and, thus, to increase the efficiency in overviewing the underlying data. Due to the reliance on the textual information associated within an image in Social Media, image search websites lack the discriminative power to deliver visually diverse search results while keeping a high precision rate amongst the retrieved images. The textual descriptions are key to retrieve relevant results for a given user query, but at the same time provide little information about the rich image content.

The main focus of this thesis is to use visual description of a landmark by choosing the most diverse pictures that best describe all the details of the queried location from community-contributed datasets. The use case is build around a tourist where a person tries to find more information about a place he/she is potentially visiting. The person has only a vague idea about the location, knowing the name of the place. He/She uses the name to learn additional facts about the place from the Internet, for instance by visiting a Wikipedia page, e.g., getting a photo, the geographical position of the place and basic descriptions. Before deciding whether this location suits her needs, the person is interested in getting a more complete visual description of the place. Many of the previous works use either the textual metadata or the visual

features in order to compute their results. Others who have used both textual features haven't explored on different clustering algorithms that ultimately led to less diverse image results.

In this work, an end-to-end framework has been built, to retrieve relevant results that are also diverse. Queries have been extended to description based "the white marble monument" along with direct queries "taj mahal". Different retrieval re-ranking and diversification strategies are used to find a balance between relevance and diversification. Different clustering techniques in order to achieve better divergent images. Different strategies are discussed to tweak some of the existing algorithms in order to decrease the run time to single scan which works better on large datasets. Extensive experiments have been conducted on the Flickr Div150cred dataset that has around 30 different locations with 300 images each and tested on the testset provided.

DEDICATION

Dedicated to my mother whose love have been like a shower of blessing in my life, professionally as well as personally

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and indebtedness to my thesis mentor Dr. Ajay Bansal for his valuable suggestions, constant supervision, timely guidance, keen interest and encouragement throughout my thesis. This work has only been possible because of the knowledge I have gained under his guidance. The lessons learnt would help me immensely in my future endeavors. I am thankful to Dr. Srividya Bansal and Dr. Michael Findler for being supportive of my research and for willing to serve on my thesis committee.

I would also like to thank Yash Garg for the discussion we had which served well for my thesis. I would also like to thank Dr Kasim Canden whose course on Multimedia and Web Databases acted as a catalyst for me to take up this thesis. I would also like to thank the Flickr team for their impressive free image datasets focused on the image captioning problem. Without that, it would have been very difficult to find a dataset with such a diverse collection of images. I would like to thank my mom, dad and sister for encouraging me and standing with me through my struggles- words don't do justice to everything you've done for me. Finally, I would like to thank Arizona State University for providing the amenities required in the successful completion of my thesis and the department of CIDSE for the constant guidance and assistance throughout my master's program and thesis.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Overview

Social media has become the most important aspect of everyday life. Nowadays, most communication is done through social media. Imagining a life without social media for example Facebook, Instagram, and Snapchat etc has become much harder. Number of active users in Facebook has increased from around 100 million in 2008 to more than 2 billion people in 2018; Instagram has almost a billion users now while it had only 90 million users five years back in 2013. The rate of increase is quite similar in other social media services such as Twitter and Snapchat from their inception. In each form of social media, retrieving results that are both relevant and divergent is always critical from user point of view. Existing retrieval technology focuses almost exclusively on the accuracy of the results (Smeulders et al. 2000), (Priyatharshini and Chitrakala 2012), (Datta et al. 2008). In the information retrieval context, a typical search involves extracting appropriate features from the query and then perform matching to the instances in the database to find similar results that are relevant to the user query. Increasingly, this technique has drawn more and more attention from the extant web search engines (e.g. Google, Yahoo!, Bing, Facebook graph search, twitter hash tags and so on). The nature of retrieval task in each case is different and also depends on the goals and intentions of the target users. More often than not the user is only vaguely aware of his/her intent. Also, each user‘s intent is different of the others, and it is better to show both relevant and diverse sets of results to maximize

the reliablity of the retrieval system. Here, users would expect to retrieve not only representative photos but also diverse results depicting the query in a comprehensive and complete manner. Another equally important aspect is that retrieval should focus on summarizing the query with a small set of images, since most of the users commonly browse only the top retrieval results. This paper mainly focuses on retrieving results that tend to give equal weightage to both precise and divergent results. The main novelty of this task is in addressing the social dimension that is reflected both in its nature (variable quality of photos and of metadata) and in the methods devised to retrieve it.

1.1.1 Motivation

The need for diversity is not limited to retrieval and there has been significant research in many applications (Dagli, Rajaram, and Huang 2006), (Khan, Drosou, and Sharaf 2013), (Ntoutsi et al. 2014). One such research is in the field of image search in social media. Over the years there have been many advances and challenges proposed to solve this by Multimediaeval.org, one such challenge is taken in this work *i.e.*, The 2014 Retrieving Diverse Social Images Task.

The importance of presenting a set of results that are at the same time relevant to the query but also exhibit diversity has been pointed out long ago in the Information Retrieval (IR) community (Marques and Furht 2002). Diversity in top results provides a more comprehensive and concise answer to the query which in turn enables faster access to the desired information and ultimately results in increased user satisfaction. Despite this fact, existing image search engines (either operating on web scale, e.g., Google Images, or within media sharing platforms, e.g., Flickr) still focus primarily

Figure 1. The first 10 results returned by Flickr - Google and our Algorithm

on relevance. As a consequence, top results usually contain many similar images and the user has to go deeper down the list of results in order to discover diverse views of the query. In addition, the deeper one goes down the list, the higher the probability to encounter irrelevant results becomes, thus impeding the discovery of diverse views. This focus on relevance is perhaps due to the limitations imposed by relying mostly on the textual modality of the images (e.g., surrounding web page text in the case of Google Images, tags and textual descriptions in the case of Flickr). Obviously, ignoring the visual content of the images limits the ability of a search engine to provide either relevant or diverse results. Figure 1 shows the first 10 results returned by Flickr (top) and Google images (middle) in response to a query about "La Madeleine" church in Paris. Both result sets are not optimal in the sense that they contain irrelevant and/or similar images. Using this dataset from 2014 Retrieving Diverse Social Images Task, our work shows how both textual metadata and visual data is used to obtain results that are both relevant and diverse. As such, we will be considering both the textual and visual data in both of the sub problems and show how each of them

is used to solve relevancy and diversity. Also, the run time performance of various methods is discussed in detail.

1.1.2 Problem Statement

In this thesis, the problem is build around a tourist use case where a person tries to find more information about a place she is potentially visiting. The person has only a vague idea about the location, knowing the name of the place. She uses the name to learn additional facts about the place from the Internet, for instance by visiting a Wikipedia1 page, e.g., getting a photo, the geographical position of the place and basic descriptions. Before deciding whether this location suits her needs, the person is interested in getting a more complete visual description of the place. As a part of the problem, the dataset has a list of photos for a certain location retrieved from Flickr2 and ranked with Flickr's default "relevance" algorithm. These results are typically noisy and redundant. The requirements of the problem is to refine these results by providing a ranked list of up to 50 photos that are considered to be both relevant and diverse representations of the query according to the definitions:

Relevance: a photo is relevant for the location if it is a common visual representation of the location, e.g., different views at different times of the day/year and under varying weather conditions, inside views, close-ups on architectural details, drawings, sketches, creative views, etc, which contain partially or entirely the target location. Photos of poor quality (e.g., severely blurred, out of focus, etc) as well as photos showing people in focus (e.g., here is me with friends in front of the monument) are not considered relevant.

Diversity: a set of photos is considered to be diverse if it depicts different visual

characteristics of the target location, e.g., different views at different times of the day/year and under varying weather conditions, inside views, close-ups on architectural details, creative views, etc, with a certain degree of complementarity, *i.e.*,, most of the perceived

Chapter 2

BACKGROUND LITERATURE

2.1 Different Learning Techniques and Their Application to Image Retrieval and Diversification

2.1.1 Image Classification

In context of image retrieval, image classification has often been used as a pre-processing step for reducing the response time to query image in large databases and improving accuracy (Haralick, Shanmugam, and Dinstein 1973). More elaborately, in image retrieval systems, a query is classified into one of category in database predicted class, subsequently, a similarity measurement step is carried out over only those images that belong to the same category as predicted for the query.

Image classification is applicable only when labeled training images are available. Domain-specific database such as medical images database(Antonie, Zaiane, and Coman 2001), remotely sensed imagery are example of databases where labeled training images are readily available. Classification methods can be partitioned into two major branches: discriminative classification approach, generative classification approach. In discriminative approach decision boundaries are estimated directly, e.g., SVM and decision trees. This approach does not need any prior information about classes. In generative approach, the density of data with in each class is estimated separately and Bayes formula is then used to compute the posterior on test data. This approach is easier to incorporate any prior information and more efficient when there

are many classes. Various algorithms that are used will be discussed in the coming sections.

2.1.1.1 Transfer Learning with CNN

In the field of computer vision, researchers have repeatedly shown the value of transfer learning — pre-training a neural network model on a known task, for instance ImageNet, and then performing fine-tuning — using the trained neural network as the basis of a new purpose-specific model(Raina et al. 2007). Transfer learning from VGG16, VGG19, ResNet-34, ResNet-50 etc. has gained in significance. The most recently developed pre-trained model is the ResNet50 model which allows for even better learning in deep networks when compared to InceptionV3 and VGG. Similar to both these architectures, a ResNet50 also is comprised of a series of convolution layers followed by fully-connected layers. We can obtain image embeddings from a ResNet-50 by taking the output of its second last Fully-connected layer. These features that are extracted can be used for classification task in our work.

Using CNNs, an approximate algorithm is used to solve the problem of nearest neighbour. There are a couple of different formulations, but the main idea is that they only need to return instances whose distance to the query point is almost that of the real nearest neighbors. Allowing for approximate solutions opens the door to randomized algorithms, that can perform an ANN (approximate NN) query in sublinear time. A Locality-Sensitive Hashing(Datar et al. 2004) scheme for the approximate nearest neighbor problem has been used to find the images that are similar to the query image. The locality sensitive hashing(LSH) reduces the computational complexity to O(log N) and thus improve the over all run time of finding similar images.

2.1.2 Image Clustering

When labeled data is not available, unsupervised clustering can be useful for speeding up the retrieval process. Image clustering is specifically applicable to web image data where meta data is also available for exploitation in addition to visual features (Wang et al. 2004),(Gao et al. 2005),(Cai et al. 2004).

Clustering techniques can be partitioned into three categories: pair-wise distance based, optimization of an overall clustering quality measure and statistical modeling. The pair-wise distance based methods, e.g., linkage clustering and spectral graph partitioning do not depend on the mathematical representation of data instance, hence they have general applicability. They are particularly useful in image retrieval because image representation may be often very complex. However, they have one disadvantage of having high computational cost because we need to compute an order of n*n pair-wise distances, where n is the size of the data. Various clustering algorithms are discussed in the coming sections

2.2 Supervised Learning Algorithms

2.2.1 Naive Bayes

Naive Bayes is a family of statistical algorithms we can make use of when doing text classification (McCallum, Nigam, et al. 1998). One of the members of that family is Multinomial Naive Bayes (MNB). One of its main advantages is that you can get really good results when data available is not much (a couple of thousand tagged samples) and computational resources are scarce.

All you need to know is that Naive Bayes is based on Bayes's Theorem, which helps us compute the conditional probabilities of occurrence of two events based on the probabilities of occurrence of each individual event. This means that any vector that represents a text will have to contain information about the probabilities of appearance of the words of the text within the texts of a given category so that the algorithm can compute the likelihood of that text's belonging to the category.

2.2.2 Support Vector Machines

Support Vector Machines (SVM) is just one out of many algorithms we can choose from when doing text classification (Colas and Brazdil 2006). Like naive bayes, SVM doesn't need much training data to start providing accurate results. Although it needs more computational resources than Naive Bayes, SVM can achieve more accurate results.

In short, SVM takes care of drawing a "line" or hyperplane that divides a space into two subspaces: one subspace that contains vectors that belong to a group and another subspace that contains vectors that do not belong to that group. Those vectors are representations of your training texts and a group is a tag you have tagged your texts with.

2.2.3 Deep Learning

Deep learning is a set of algorithms and techniques inspired by how the human brain works. Text classification has benefited from the recent resurgence of deep learning architectures due to their potential to reach high accuracy with less need of engineered

features. The two main deep learning architectures used in text classification are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

On the one hand, deep learning algorithms require much more training data than traditional machine learning algorithms, *i.e.*, at least millions of tagged examples. On the other hand, traditional machine learning algorithms such as SVM and NB reach a certain threshold where adding more training data doesn't improve their accuracy. In contrast, deep learning classifiers continue to get better the more data you feed them with

2.2.4 Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary) (Genkin, Lewis, and Madigan 2007). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

2.2.5 Word2vec and Logistic Regression

Word2vec, like doc2vec, belongs to the text preprocessing phase. Specifically, to the part that transforms a text into a row of numbers. Word2vec is a type of mapping that allows words with similar meaning to have similar vector representation. The idea behind Word2vec is rather simple: we want to use the surrounding words to represent the target words with a Neural Network whose hidden layer encodes the word representation. BOW based approaches that includes averaging, summation,

weighted addition. The common way is to average the two word vectors. Therefore, we will follow the most common way in our method.

2.2.6 Doc2vec and Logistic Regression

The same idea of word2vec can be extended to documents where instead of learning feature representations for words, we learn it for sentences or documents. To get a general idea of a word2vec, think of it as a mathematical average of the word vector representations of all the words in the document. Doc2Vec extends the idea of word2vec, however words can only capture so much, there are times when we need relationships between documents and not just words.

The way to train doc2vec model for our Stack Overflow questions and tags data is very similar with when we train Multi-Class Text Classification with Doc2vec and Logistic Regression.

2.3 Clustering Algorithms

Stated below are the various clustering algorithms used in our thesis:

2.3.1 Graph Based Clustering

As in multidimensional scaling, which embeds a given set of objects into a metric vector space using the a priori knowledge about the distances among them, graph-based clustering techniques also embed the objects into another platform for clustering. Unlike MDS, however, these techniques embed the data into a graph (instead of a

vector space), which is then analyzed for identifying clusters. The general outline of the graph-based clustering methods is as follows:

- Compute the similarity/distance between all object pairs
- Compute a threshold if not already given
- Create a graph that represents each object with a node and each pair whose similarity is abov the threshold (or distance less than the threshold) with an edge
- Analyze the resulting graph to identify clusters

Below are the two graph based clustering techniques used in this thesis work:

2.3.1.1 Spectral Graph Partitioning

The problem of partitioning a graph into clusters using cliques is an NP-hard problem. An alternative is to rely on spectral clustering, where the eigenvectors of the adjacency matrix (describing the pairwise similarities of the nodes) or the Laplacian matrix (describing the second-order connectivity) of the graph are used for identifying clusters [Boppana, 1987; Fiedler, 1973; Kannan et al., 2000; McSherry, 2001; Pothen et al., 1990; Schaeffer, 2007; Spielman and Hua Teng, 1996]. In the former case, the eigenvalues indicate the path capacity of the graph [Harary and Schwenk, 1979]; in the latter case, they indicate its algebraic connectivity [Chung, 1997; Fiedler, 1973]. An advantage of the spectral clustering algorithms over clique-based approaches is that, using randomized algorithms, such as [Drineas et al., 1999] and [Frieze et al., 1998], spectral clustering can be implemented in nearly linear time. In our thesis we have used angular clustering. An angular spectral clustering algorithm first finds the k left singular vectors, u1, . . . , uk with the highest singular values, y1, . . . , yk.

Each of these k singular vectors corresponds to a cluster. Let the clustering matrix, C, be such that the jth column (corresponding to the jth cluster) is equal to yjuj. Then, the node ni of the graph is placed in cluster j if the largest entry in the ith row is C[i, j] [Kannan et al., 2000]. Note that, essentially, one assigns each node to the cluster whose singular vector has the smallest angle from the adjacency vector of the node.

In this thesis, as discussed in the previous step, relevant images are obtained for a given query using the textual metadata. Using the top 100 images that are retrieved, we use K nearest neighbours in order and generate a graph with edges being the similarity/distance between each of the images. This graph then broken down in N different clusters. We then go on and select an image for each of the cluster. Based on the number of images to be shown to the user, we decide the number of images that needs to chosen from each of the cluster by comparing their relative sizes.

2.3.1.2 Metis - Multilevel Graph based Clustering Algorithm

The graph partitioning problem is NP-complete. However, many algorithms have been developed that find a reasonably good partition. Spectral partitioning methods are known to produce good partitions for a wide class of problems, and they are used quite extensively. However, these methods are very expensive since they require the computation of the eigenvector corresponding to the second smallest eigenvalue (Fiedler vector). Execution time of the spectral methods can be reduced if computation of the Fiedler vector is done by using a multilevel algorithm.

Formally, a multilevel graph bisection algorithm works as follows: consider a weighted graph G0 = (V0;E0), with weights both on vertices and edges. A multilevel graph bisection algorithm consists of the following three phases. Coarsening phase.

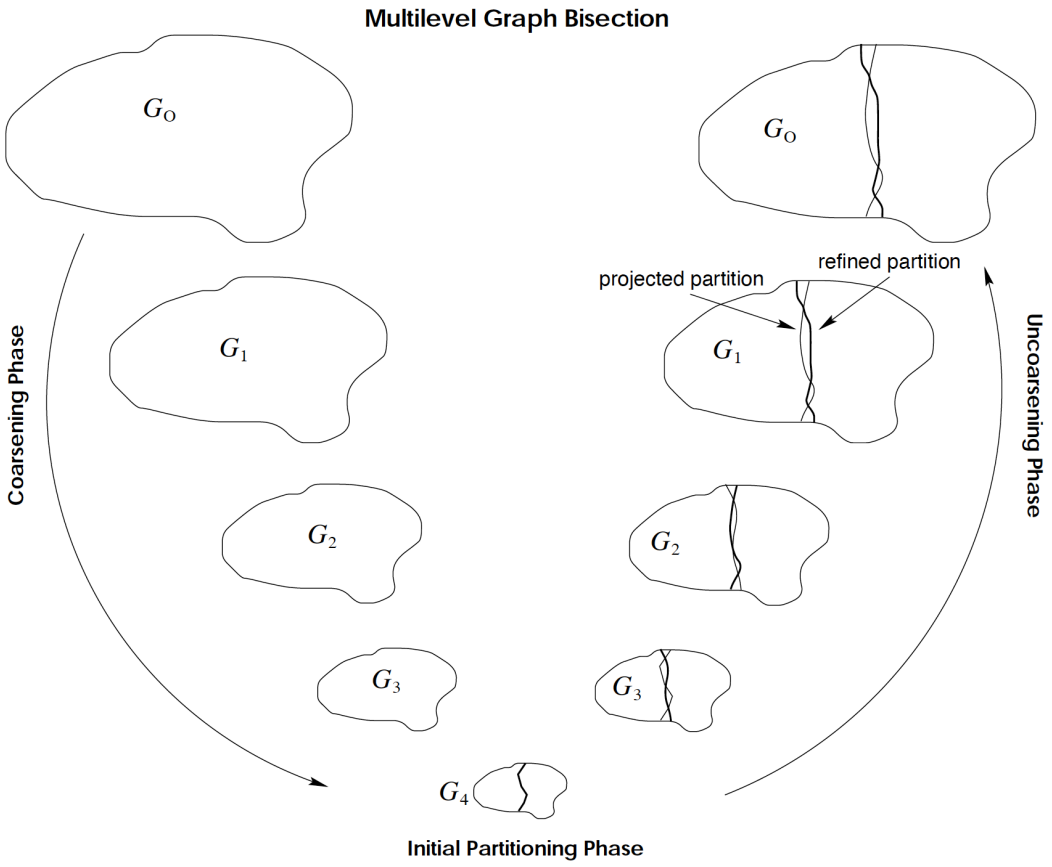

Figure 2. The various phases of the multilevel graph bisection

The graph G0 is transformed into a sequence of smaller graphs G1, G2, ... Gm such that V0 > V1 > V2 > ..... > Vm.

- Partitioning phase. A 2-way partition Pm of the graph Gm = (V;Em) is computed that partitions Vm into two parts, each containing half the vertices of G0.
- Uncoarsening phase. The partition Pm of Gm is projected back to G0 by going through intermediate partitions Pm, P(m-1), ... P1, P0.
- Coarsening phase. During the coarsening phase, a sequence of smaller graphs,

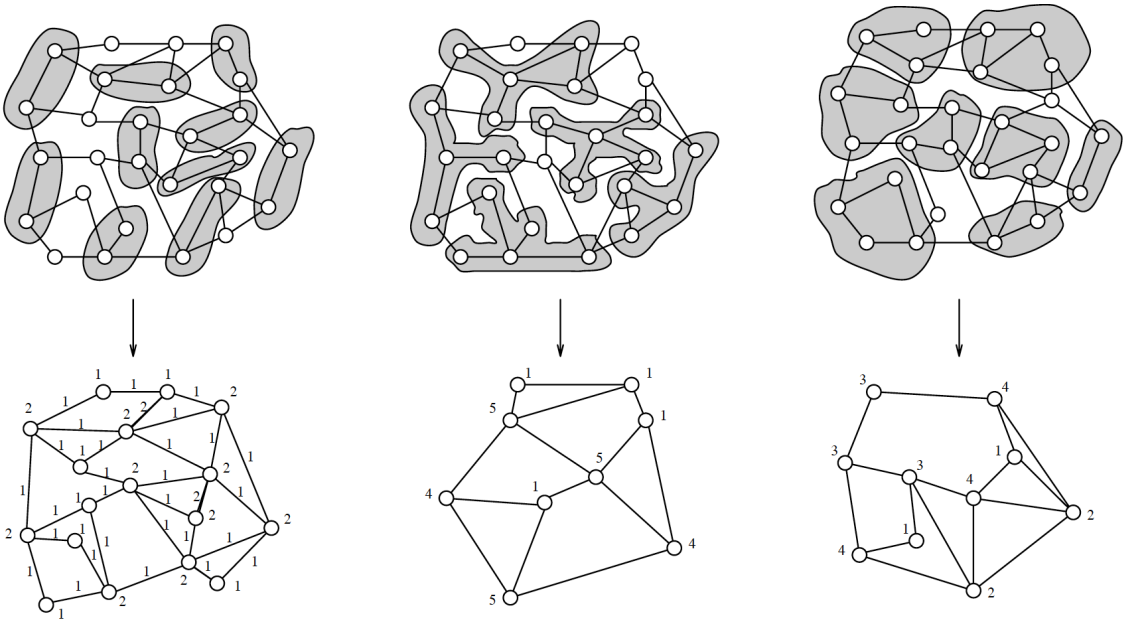

Figure 3. Different ways to coarsen a graph.

each with fewer vertices, is constructed. Graph coarsening can be achieved in various ways. Some possibilities are shown in Figure below.

The processes followed is the same as that of spectral clustering. Initially we created a K nearest neighbour graph. Using metis library, we are able to give this graph as an input and then partition it.

2.3.2 Iterative Clustering

The graph-based clustering algorithms we discussed in the previous section all have at least O(N2) initial cost, where N is the number of objects in the database, because they require distance or similarity values to be computed for all pairs of objects. For large databases, computing pairwise scores may simply be infeasible. A second category of clustering methods, commonly referred to as the iterative clustering

methods, try to avoid the quadratic time complexity and reduce the execution time to O(N). It is easy to see that the cost of iterative algorithms is in general O(kN), where k is the number of resulting clusters: the main loop goes over each object once and compares this object to all clusters created so far to pick the most suitable cluster. Thus, the total amount of work performed per object is at most O(k), and the cost of the algorithm is linear in the number, N, of objects.

2.3.2.1 K Means Clustering Algorithm

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. To process the learning data, the K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.

It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

2.4 Image Features

In literature, a number of image features have been used ,e.g., Global Color Naming Histogram, Global Histogram of Oriented Gradients, Global Color Moments on HSV Color Space, Global Locally Binary Patterns on gray scale, Global Color Structure Descriptor, Global Statistics on Gray Level Run Length Matrix etc are used.

2.4.1 Motivation Behind Using Local Feature Vectors of Images

The global feature is sensitive to change due to perceptive distortion, occlusions, illumination variations and clutter. It may also be redundant, contain misleading information and fail to capture finer semantic details of an image (Bosch, Muñoz, and Martı 2007). Local features are powerful image descriptors and provide better image representation when compared to global features (Verma 2014).

2.4.2 Extraction of Local Features

Local features can be extracted from an image using following three approaches:

- Image is segmented into regions and from each region a feature vector is extracted.
- Salient point in images are detected and a feature vector is extracted from a region around each of these salient points.
- Image is partitioned into fixed size blocks and from each block a feature vector is extracted.

2.4.3 Using CNN for Feature Extraction

CNN is currently the state-of-the-art architecture for solving visual recognition problems(Wu, Shen, and Van Den Hengel 2019). The core problem solved by them is the classification problem where objects in images are classified according to their class. A set of architectures have been trained on immensely large datasets of images which are the current top-notch architectures for classifying images. In our work, we

used ResNet50 in order to extract the image features. This features are used and compared with the local descriptors available.

Chapter 3

RELATED WORK

Many of the works address relevant results. It uses the textual data in order to do it. When you have results that are almost the same, the uses might not be happy about the results that are generated. There results are not also well rounded. Firstly, many of the works used textual data to get the relevant results. These are used to tackle the query needs that are not really clear(Dang and Croft 2012). For most of the results, they have one main goal *i.e.*, improve the results in retrieval scenarios. In any case, there may be situations where this probably won't be valid. There can be occasions where a query can be comprised of various sub-queries and we need our consideration on these. These sub-queries can include many sub queries such as for instance sub-topics, e.g., bikes that are of different producers, animals are of different species, object that are having many different shapes, photos with filters, points of interest can be photographed from different angles and so on. Therefore, one should consider equally the diversification in a retrieval scenario.

Query like discussed above contains many intents or interpretations. We call these the subtopics. All of these needs to be added inorder to improve on the query diversity(Dang and Croft 2012). By enlarging the pool of potential outcomes, one can improve the probability of the retrieval framework to give the user with data required and in this manner to build its proficiency. By widening the pool of possible results, one can increase the likelihood of the retrieval system to provide the user with information needed and thus to increase its efficiency. For instance, in user recommender systems, users will find satisfactory results much faster if the diversity

of the results is higher (McGinty and Smyth 2003). In increasing the diversity, the user needs to take care of the relevant search results i.e relevance score shouldn't fall due to the increase in the amount of diversity. This is one of the issue we will look to resolve in our work.

Image retrieval results can be categorized into two different type. Here is an example work that shows the same(Deng and Fan 2014). Now as stated, the algorithm that we use for ranking the results will give us a set of results (S) that are assumed to be relevant to the user's query. Next, we need to find a subset of results (R). There results are relevant and now diverse, *i.e.*,, in contrast to the other elements from the set R. The final result to to again make a balance - a balance between relevance and diversity which in general is antinomic - increase relevance will decrease diversity and vice versa. Now you don't want to do too much of diversification. This might result in not getting the relevant items and getting images that are relevant for other queries. On the other side, we have the problem of duplicates. Now, our work need to ensure that there is always a balance - less duplicates and relevant images.

The most mainstream content diverification and enhancement methods investigate Greedy advancement arrangements that assemble the outcomes in a steady manner (a review is presented in (Deng and Fan 2014)). For instance, (Santos, Macdonald, and Ounis 2015) is one work where a probabilistic model has been used. Relevance is achieved with standard ranking while diversity is performed through categorization according to a certain variant results. Now the methods are analyzed to produce a set of results that cover this variance of the query. (Vee et al. 2008) uses a Greedy algorithm to compute a result set where diversification is achieved according to the document frequency in the data collection. Another example is the approach in (Zhu et al. 2007) that uses absorbing Markov chain Random walks to re-rank documents.

A document that was already ranked becomes an absorbing state, dragging down the importance of similar unranked states. Transposed to multimedia items and more specifically in the context of social media, the diversification receives a new dimension by addressing multi-modal (visual-text) and spatio-temporal information (video). Due to the heterogeneous nature of modalities, multimedia information is more complex and difficult to handle than text data. Assessing the similarity between multimodal entities has been one of the main research concerns in the community for many years. Common approaches are attempting to simplify the task by transposing the rich visual-text information intomore simple (numeric) representations such as using content descriptors and fusion schemes. Diversification is then carried out in these multi-dimensional feature spaces with strategies that mainly involve machine learning (e.g., clustering).

Now there are other works that are looked into for diversification. Not all the datasets can use textual data for diversity. Now we can look into other approaches used for diversification. For instance, (Van Leuken et al. 2009) is one work that uses visual diversification. Now there are clustering techniques that are used along with a dynamic weighting function. This is at best able to capture the discriminative aspects of some of the image results. How do you get the results then. Now various images will be selected from each of the clusters. (Deselaers et al. 2009) is one such work that is used to jointly optimizes the diversity and the relevance of the images in the retrieval ranking using techniques inspired by Dynamic Programming algorithms. Another work, (Taneva, Kacimi, and Weikum 2010) is another work that intends to populate a database with high exactness also, various photographs of various elements by revaluating social realities about facts about the entities. In this work, the authors use a model parameter that is estimated from a small set of training entities. Visual

similarity is exploited using the classic Scale-Invariant Feature Transform (SIFT). (Rudinac, Hanjalic, and Larson 2013) addresses the problem of image diversification in the context of automatic visual summarization of geographic areas and exploits user-contributed images and related explicit and implicit metadata collected from popular content-sharing websites. The approach is based on a Random walk scheme with restarts over a graph that models relations between images, visual features, associated text, as well as the information on the up loader and commentators.

Even though there are some works that used a clustering algorithm, we felt that none of then experimented with different clustering algorithms. Although it is know that spectral clustering perform well compared to other techniques, is it computationally expensive. In order to generalize for different datasets, cluster ensembles (Boongoen and Iam-On 2018) have been widely studied and will be one of the technique we will be using inorder to build a model for variable datasets.

Chapter 4

DATA PREPROCESSING AND FEATURE EXTRACTION

4.1 Data Filtering

Data filtering is one of the major step that is needed in out task. Since the dataset is unique and not one of the most popular datasets the machine leaning and data scientists are working on, there are many of the outlines that needs to be filtered before. Going through the dataset manually, below are some of the major outlines that have been observed.

- Many of the images that are retrieved for a given query contains images with faces (i.e. people occupying major part of the image)
- Many of the images are far from the point of interest. This might negate the performance of recommendation system by retrieving results that are far from away from the point of interest.
- Some of the images have very few views based on the flick data provided
- Some of the images have description that is more than 2000 characters long.

Data Filtering is based on 3 subdivisions: 1) Filtering based on distance of image from the actual geolocation 2) Filtering based on the presence of faces in images. 3) Filtering based on number of views

4.1.1 Filtering Face/People in Images

The top results from flickr results contain lot of noise. Below(Figure 4) is an example figure that shows results retrieved from a image hosting website - Instagram.

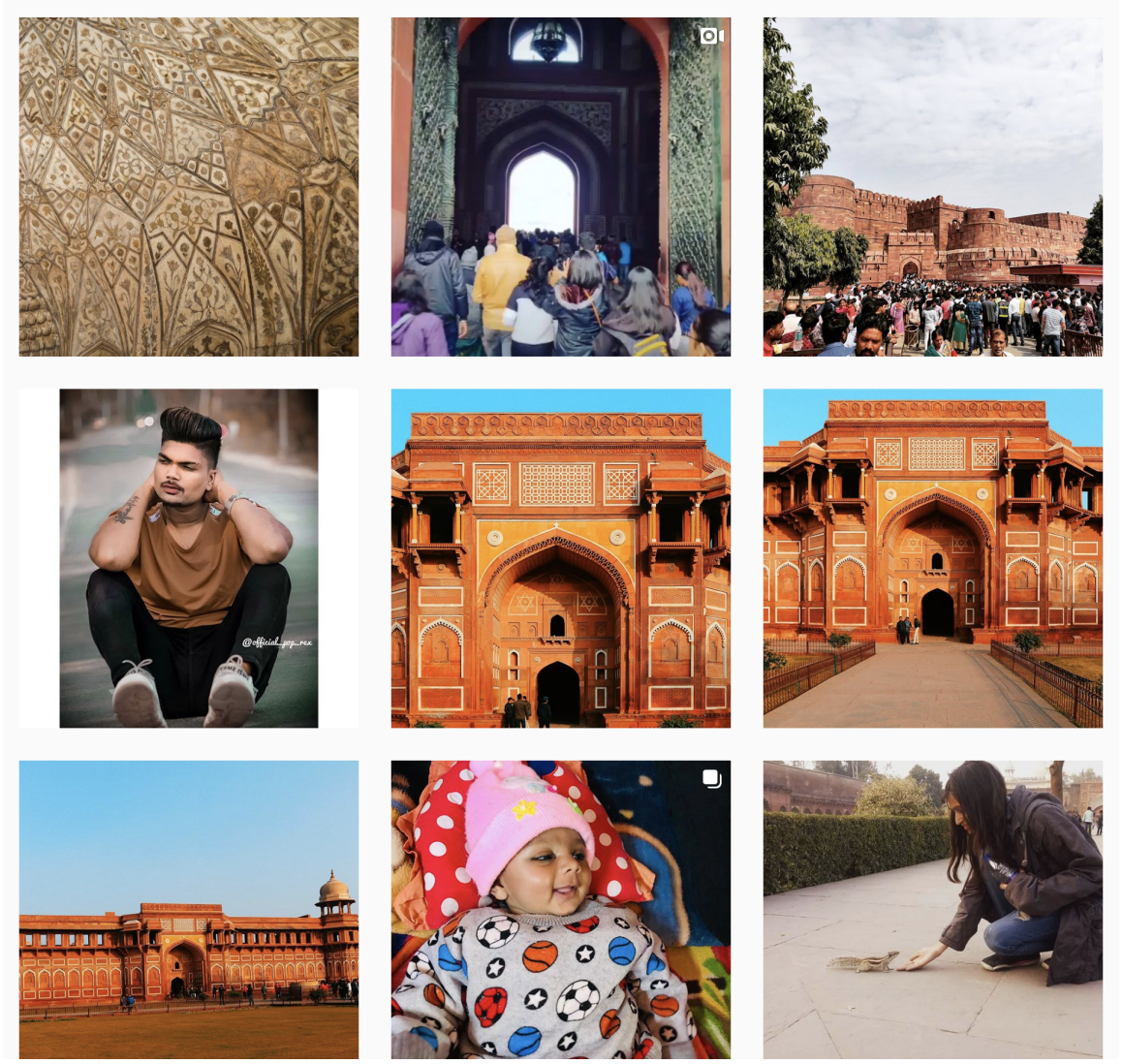

Figure 4. Top results for query: 'Agra Fort' showing results containing noise

In the problem we are trying to solve, as we need to retrieve very few results (maximum of 50) in our end results, the filter is a restriction on the presence of faces in images. We use the standard OpenCV algorithm to perform face detection and we eliminate images having a face coverage ratio higher than 0.4.

In order to detect faces, we are using OpenCV. The following figure (Figure 5) above shows the snippet of code used in our work.

```
face_cascade = cv2.CascadeClassifier('../devset/haarcascade_frontalface_default.xml')
for photo in root:
    photoCheck = "../devset/img/" + poi_name + "/" + photo.attrib['id'] + '.jpg'
    img = cv2.imread(photoCheck)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)
    if(type(faces)==tuple):
    photos.append(photo.attrib)
```

Figure 5. Code snippet where all the images in our database are filtered using OpenCV's Face Cascade

Figure 6 shows that results for query 'Hearst Castle'. Of the results, image 6092892474 contains human face that is occupying more than 40 percent of the image. These images, when considered in the final results might not much of an use to the end users. As we want to depict the given query in a few images (10-20), we chose remove images that has face (occupied with people) as we consider these as noise in the dataset. Also, Image 6460230437 contains many people but the none of them have occupied more than 40 percent of the space. So, results like these are included in one next step.

4.1.2 Filtering Based on Distance of the Image from the Query Location

For a given location query, it is observed that there are many images that didn't reflect the geolocation. Example: When a user searches for a particular location, the results can contain images of the exact location *i.e.,*, the buildings at the location, towers, gardens nearby etc or iamges of nearby places. In order to improve on the user experience, we eliminated geotagged images that have a distance from the point of interest higher than 5 kms. Since the Earth does not exactly follow one of the geometric shapes, calculating the distance on its surface is one of the important challenges. Haversine and Vincenty formulas are the two major methods used for

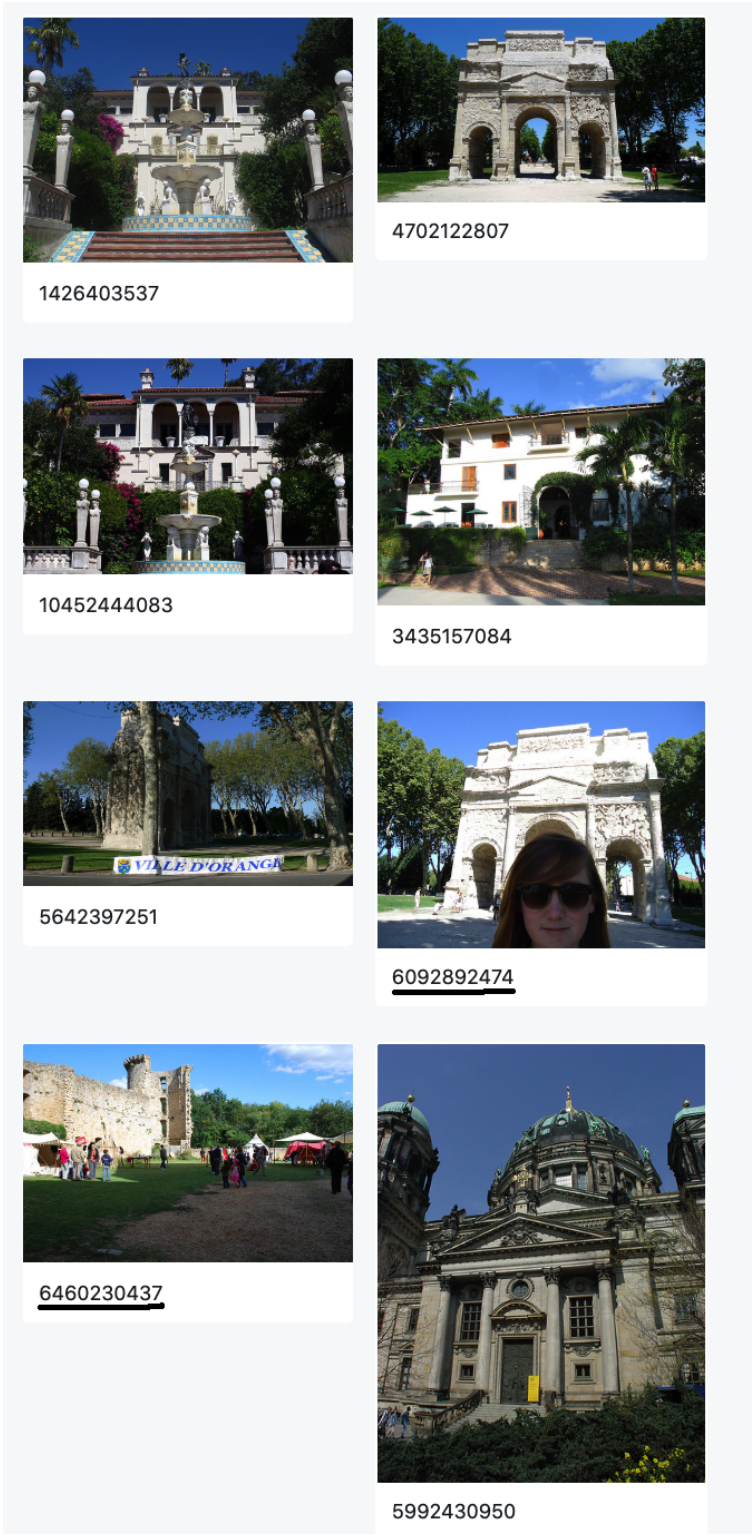

Figure 6. Results for query 'Hearst Castle' before filtering

```
def haversine(row,gps):
    # distance between latitudes
    # and longitudes
    lat1, lon1 = gps[row["poiName"]]
    lat2, lon2 = float(row["latitude"]), float(row["longitude"])
    dLat = (lat2 - lat1) * math.pi / 180.0
    dLon = (lon2 - lon1) * math.pi / 180.0

    # convert to radians
    lat1 = (lat1) * math.pi / 180.0
    lat2 = (lat2) * math.pi / 180.0

    # apply formulae
    a = (pow(math.sin(dLat / 2), 2) +
         pow(math.sin(dLon / 2), 2) *
         math.cos(lat1) * math.cos(lat2));
    rad = 6371
    c = 2 * math.asin(math.sqrt(a))
    return rad * c
```

Figure 7. Code for Haversine distance

calculating distances on a sphere and elliptic shapes, respectively. Since the Earth is neither a perfect sphere nor ellipse, using these formulas gives approximate results about the real distances. In our work, we have used Haversine (Chopde and Nichat 2013) in order to get the appropriate results.

4.1.3 Filter Blurred Images

The dataset we are using has been provided by Flickr and the main novelty the problem we are addressing is the social dimension that is reflected both in its nature (variable quality of photos and of metadata) and in the methods devised to retrieve it. Going through the list of images provided, we can see that there are images which are

blurred, filled with smog (tampered by the weather conditions) or in is not properly focused (Figure 8). All these images are considered as outliers as these images will not be useful for an end user to understand the location he is searching for fully well.

The first method considered in our work is would be computing the Fast Fourier Transform of the image and then examining the distribution of low and high frequencies — if there are a low amount of high frequencies, then the image can be considered blurry. However, defining what is a low number of high frequencies and what is a high number of high frequencies can be quite problematic, often leading to sub-par results. Also, the method might work on large data sets as the algorithm will not complete is a single pass. Instead, our work an implementation, a variation of the Laplacian by Pech-Pacheco. In this method, we take a single channel of an image (presumably grayscale) and convolve it with the following 3 x 3 laplacian kernal and then take the variance (*i.e.*, standard deviation squared) of the response. If the variance falls below a pre-defined threshold, then the image is considered blurry; otherwise, the image is not blurry. The reason this method works is due to the definition of the Laplacian operator itself, which is used to measure the 2nd derivative of an image. The Laplacian highlights regions of an image containing rapid intensity changes. Also, Laplacian is often used for edge detection. The assumption here is that if an image contains high variance then there is a wide spread of responses, both edge-like and non-edge like, representative of a normal, in-focus image. But if there is very low variance, then there is a tiny spread of responses, indicating there are very little edges in the image. In this way, we are able to filter results that are blurred.

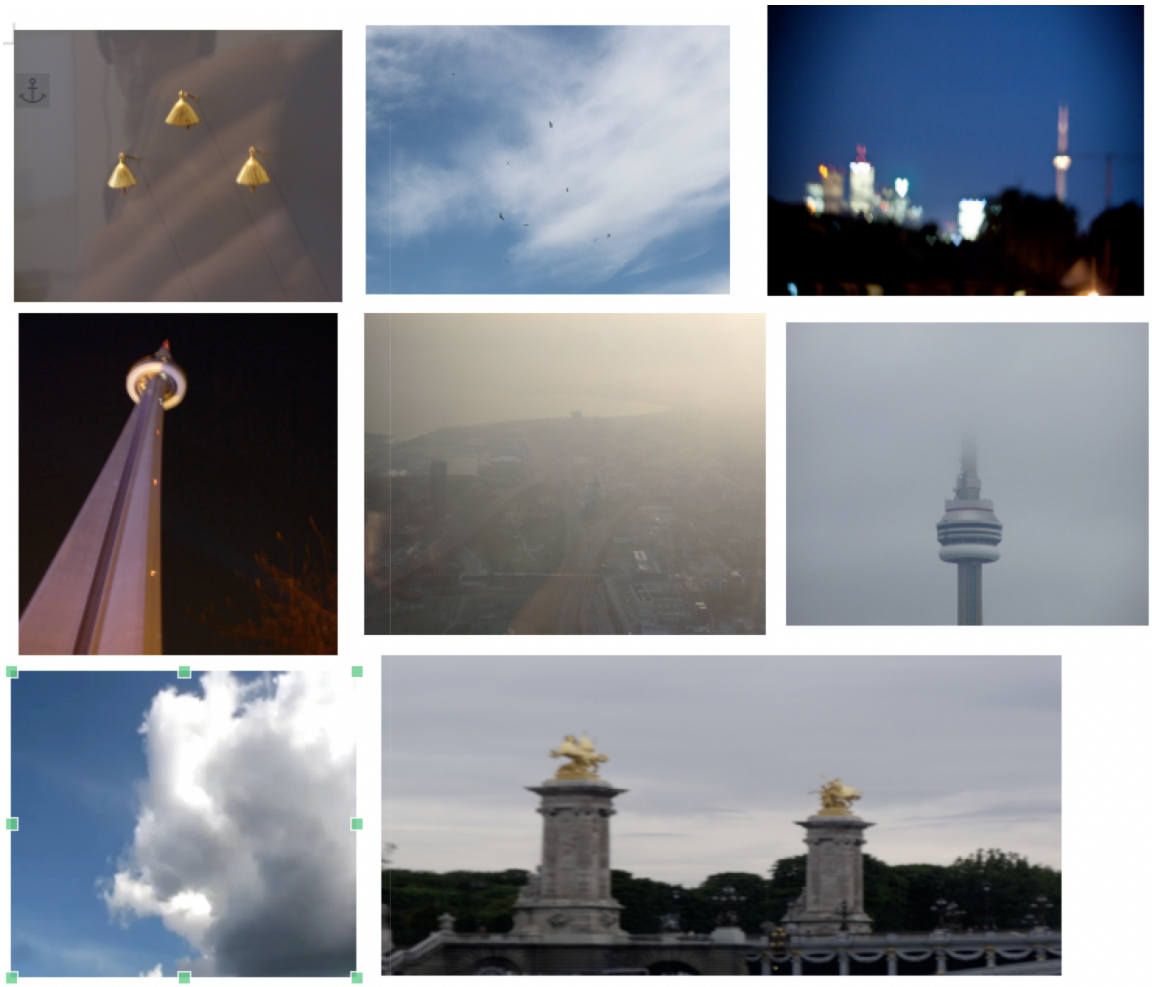

Figure 8. Some of the blurred images

4.1.4 Data Statistics

The data contains devset and testset. Devset contains 30 different locations around 300 images each. Testset data contains 15 different location around 300 images.

Table 1. Number of Images for each location before filtering

Tourist Location	Number of Images
Agra Fort	296
Albert Memorial	299
Altes Museum	296
Amiens Cathedral	296
Angel of the North	299
Angkor Wat	296
Ara Pacis	297
Arc De Triomphe	297
Aztec Ruins	300
Berlin Cathedral	297
Big Ben	298
Bok Tower Gardens	299
Brandenburg Gate	300
Cabrillo	293
Casa Batllo	298
Casa Rosada	299
Castillo De San Marcos	300
Chartres Cathedral	299
Chichen Itza	297
Christ The Redeemer Rio	289
Civic Center SF	298
CN Tower	297
Cologne Cathedral	299
Colosseum	300
Hearst Castle	298
La Madeleine	296
Montezuma Castle	299
Neues Museum	296

After all the filtering the number of images per location came up to 258. On average 32 images out of 300 images have been filtered in the initial steps.

Table 2. Number of Images for each location after filtering facial results

Tourist Location	Number of Images
Acropolis Athens	269
Agra Fort	277
Albert Memorial	281
Altes Museum	272
Amiens Cathedral	261
Angel of the North	268
Angkor Wat	284
Ara Pacis	282
Arc De Triomphe	289
Aztec Ruins	272
Berlin Cathedral	269
Big Ben	281
Bok Tower Gardens	273
Brandenburg Gate	266
Cabrillo	258
Casa Batllo	282
Casa Rosada	270
Castillo De San Marcos	259
Chartres Cathedral	291
Chichen Itza	284
Christ The Redeemer Rio	255
Civic Center SF	266
CN Tower	275
Cologne Cathedral	264
Colosseum	283
Hearst Castle	277
La Madeleine	264
Montezuma Castle	254
Neues Museum	262

4.2 Feature Extraction

The dataset contains few of the visual descriptors that are global in nature *i.e.*, they don't clearly capture the local minute details that are needed for image discrimination. Below are the few of the features that are already available in the dataset.

Table 3. Number of Images for each location after filtering facial results and blurred images

Tourist Location	Number of Images
Acropolis Athens	264
Agra Fort	274
Albert Memorial	277
Altes Museum	271
Amiens Cathedral	259
Angel of the North	266
Angkor Wat	283
Ara Pacis	280
Arc De Triomphe	285
Aztec Ruins	272
Berlin Cathedral	268
Big Ben	279
Bok Tower Gardens	272
Brandenburg Gate	266
Cabrillo	257
Casa Batllo	279
Casa Rosada	270
Castillo De San Marcos	259
Chartres Cathedral	290
Chichen Itza	283
Christ The Redeemer Rio	255
Civic Center SF	266
CN Tower	272
Cologne Cathedral	264
Colosseum	282
Hearst Castle	276
La Madeleine	263
Montezuma Castle	254
Neues Museum	261

- Global Color Histogram
- Global Color Moments
- Global Color Structure Descriptor
- Global Stats on Gray Level Run Length Matrix

Table 4. Number of Images for each location after filtering facial results, blurred images ad distant images

Tourist Location	Number of Images
Acropolis Athens	258
Agra Fort	261
Albert Memorial	258
Altes Museum	251
Amiens Cathedral	249
Angel of the North	248
Angkor Wat	269
Ara Pacis	271
Arc De Triomphe	271
Aztec Ruins	268
Berlin Cathedral	258
Big Ben	267
Bok Tower Gardens	259
Brandenburg Gate	257
Cabrillo	242
Casa Batllo	261
Casa Rosada	258
Castillo De San Marcos	243
Chartres Cathedral	281
Chichen Itza	267
Christ The Redeemer Rio	246
Civic Center SF	251
CN Tower	259
Cologne Cathedral	248
Colosseum	276
Hearst Castle	271
La Madeleine	245
Montezuma Castle	249
Neues Museum	252

- Color Moments 3*3
- Local Binary Patterns

Few of the features are extracted that can additionally complement the global features described above. The features are extracted using openCV library along with

Matlab predefined libraries. These features take the local patterns into consideration giving more importance to the granular details.

- Local Color Histograms
- Histogram of Oriental Gradients
- Dense SIFT
- Sparse SIFT
- Tiny Images

4.3 Data Sample

A datasample(Figure 9) contains an xml files of each individual user. Each user xml file contains list of photos taken by the user. Each image has attributes like the ones shown below:

- Data Taken
- Photo ID
- Tags
- Title
- Views
- Description
- Latitude
- Longitude

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<metadata user="7704455@N02">
  <credibilityDescriptors>
    <visualScore>0.707386884874176</visualScore>
    <faceProportion>0.047</faceProportion>
    <tagSpecificity>0.444228630950038</tagSpecificity>
    <locationSimilarity>0.566422982364553</locationSimilarity>
    <photoCount>6086</photoCount>
    <uniqueTags>0.07504264403906954</uniqueTags>
    <uploadFrequency>536.7464650695256</uploadFrequency>
    <bulkProportion>0.24646730200460074</bulkProportion>
  </credibilityDescriptors>
  <photos>
    <photo date_taken="2013-01-26 15:35:41" id="8416705667" tags="" title="2.000.000 de
visualizações / 2,000,000 views" url_b="http://farm9.static.flickr.com/
8192/8416705667_c159d2fc19_b.jpg" userid="7704455@N02" views="143" />
    <photo date_taken="2012-08-24 18:31:23" id="8132259895" latitude="45.428963"
longitude="-72.617495" tags="park parque canada tree forest quebec deer québec floresta
árvore parc canadá yamaska veado" title="Parque Yamaska" url_b="http://
farm9.static.flickr.com/8472/8132259895_840b2f2680_b.jpg" userid="7704455@N02"
views="163" />
    <photo date_taken="2012-08-24 17:32:46" id="8132285490" latitude="45.431105"
longitude="-72.621545" tags="park parque canada grass quebec grama skunk parc striped
canadá gambá yamaska stripedskunk cangambá" title="Parque Yamaska" url_b="http://
farm9.static.flickr.com/8052/8132285490_4dca792a32_b.jpg" userid="7704455@N02"
views="201" />
    <photo date_taken="2012-08-24 17:31:56" id="8132285098" latitude="45.431111"
longitude="-72.62152" tags="park parque canada grass quebec grama stripped skunk parc
canadá gambá yamaska cangambá strippedskunk" title="Parque Yamaska" url_b="http://
farm9.static.flickr.com/8323/8132285098_8e8d908d1b_b.jpg" userid="7704455@N02"
views="167" />
    <photo date_taken="2012-08-24 15:44:58" id="8132258823" latitude="45.423027"
longitude="-72.616889" tags="park blue parque lake azul landscape lago path parc caminho
trilha yamaska" title="Parque Yamaska" url_b="http://farm9.static.flickr.com/
8327/8132258823_bc64c340a4_b.jpg" userid="7704455@N02" views="266" />
    <photo date_taken="2012-08-24 12:58:43" id="8132284358" latitude="45.427305"
longitude="-72.61607" tags="park parque people lake praia beach grass bike bicycle table
lago gente seagull bicicleta canoe grama parc mesa canoa gaivota yamaska" title="Parque
Yamaska" url_b="http://farm9.static.flickr.com/8471/8132284358_2bbde8b446_b.jpg"
userid="7704455@N02" views="228" />
    <photo date_taken="2012-08-23 12:32:42" id="8132283948" latitude="45.432575"
longitude="-73.300645" tags="" title="Desenho de crianças" url_b="http://
farm9.static.flickr.com/8473/8132283948_f852cdf76a_b.jpg" userid="7704455@N02"
views="60" />
    <photo date_taken="2012-08-22 21:12:02" id="8132257707" latitude="45.508586"
longitude="-73.553389" tags="" title="Prefeitura de Montreal" url_b="http://
```

Figure 9. XML files showing the list of images and attributes of each image of user '7704455@N02'

Chapter 5

METHODOLOGY

Initial approach was to use the complete set of images to find the relevant and divergent results using K-Nearest Neighbours. This approach was computationally very complex and didn't produce good results. To illustrate it further, we formed a dense graph connecting 8923 images as our nodes and the distance (or similarity) amongst them as the edge distance. On different runs, we used a KNN and an edge cut clustering algorithms to diversify the results. This led to two different problems:

- Not all the images are relevant in the sub groups (clusters) formed.
- A dense graph is not memory efficient and the run time complexity is exponential

Next, we tried to create a sparse graph (connecting only top nearest images for each of the images) but even this didn't solve us the relevant results problem. In order to get the better results and improve on the time complexity of the problem, we have to subdivide our problem statement.

The 2 sub problems are as follows:

- Getting most Relevant Results by re-ranking the results
- Getting Divergent Results

5.1 Most Relevant Search Results

In order to diversify our results, we first need to find the list of most relevant results for a given query. For our results, we will be using content based filtering in order achieve the desired results.

Relevant results can we retrieved using

- Textual Metadata - Using the title, tags and description provided for each image
- Visual Metadata - Using the visual data (edge detection, object detection and other features from CNNs)

5.1.1 Relevant Search Results using Textual Metadata

Textual metadata consists of the image title, tags and description. Before looking into the image retrieval methods, one major problem that we faced in our textual data is missing data. Of the 8923 images, there are 17 images in the dataset without title, 634 images without tags and 1496 images without description. Of these, the 17 images without title also doesn't contain the tags or the description.

In our thesis, we have evaluated 2 different types of queries:

- Landmark Query - Example: Agra Fort, Eiffel Tower etc
- Description based query - Example: The red sandstone fort build by mughals, a tall iron tower in Paris

The landmarks queries are used to compare our work with the other papers as it is the standard followed in the competition. The description based query is used for a

comparative study and see how our methodology is performing on variable queries that are not straightforward. Not much of the work is done in order to improve them.

The relevant result problem can be treated as both supervised and unsupervised. In the computation "MediaEval Challenge", the problem is treated as supervised. Our final evaluation is also based on this method - treating it as a supervised study. What does supervised means? In the dataset given, we are already provided with the class each of the image belongs to. There class is the name of the landmark (Example: Agra Fort, Acropolis Athens, Golden Gate Bridge etc). We are treating both of these problems in a similar fashion *i.e.*, retrieving the relevant results based on similarity ranking for a give query. In order to improve on the net results, we need to do a class evaluation in the supervised problem i.e class evaluation based on the query followed by ranking the results based on the similarity measure. This can be a bottleneck in a supervised learning algorithm. What if an algorithm is only 80 percent accurate. Then there is 20 percent chance that we are choosing the right class. This can be a very huge mistake in our final results. The problem of increasing the diversity is irrelevant when you are not able to find the right set of images. Now what if the accuracy of our prediction is 98 percent instead of 80 percent. Yes, we have got a very strong classification algorithm that predicts the class correctly 98 times out of 100. This can still be a bottleneck. What if the query that is predicted falls in the other 2 percent of the results. Again, the same out problem. How do you avert this bottleneck? In order to increase our chances of attaining the right result, we are using a Ranking based ensemble, which increases the probability of getting a right result. This method is taken after experiment classification algorithms ran show that the models are able to predict the class rightly 92 - 98 percent of the times. Table 6 and 7 illustrate how this is done. Figure 14 depicts the system.

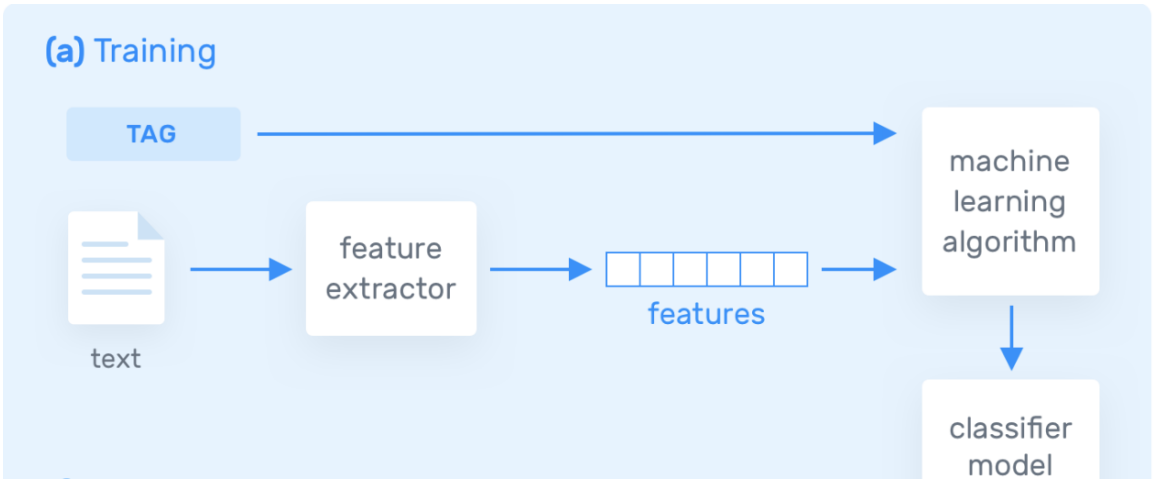

Figure 10. Training Data using ML. Source: https://monkeylearn.com/text-classification

The first step towards training a classifier with machine learning is feature extraction: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is bag of words, where a vector represents the frequency of a word in a predefined dictionary of words. For example, if we have defined our dictionary to have the following words This, is, the, not, place, in, london, and we wanted to vectorize the text "This is place", we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0). Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. sports, politics) to produce a classification model

Once it's trained with enough training samples, the machine learning model can begin to make accurate predictions. The same feature extractor is used to transform unseen text to feature sets which can be fed into the classification model to get predictions on tags (e.g. sports, politics):

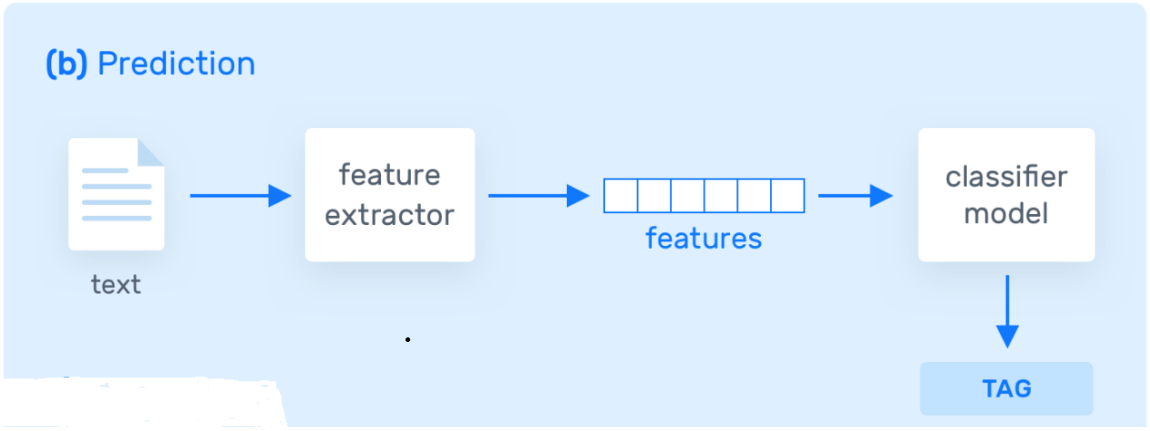

Figure 11. Predicting after training model. Source: https://monkeylearn.com/text-classification

Text classification with machine learning is usually much more accurate than human-crafted rule systems, especially on complex classification tasks. Also, classifiers with machine learning are easier to maintain and you can always tag new examples to learn new tasks. In our work, we have used different Machine Learning algorithms in order to test the performance of textual descriptors - metadata. We will be using the supervised algorithms that are discussed in our background literature.

5.1.1.1 Image Similarity with Cosine, Glove and Wiki Description

Since the images are collected from users uploading images into Flickr, many of the images that are uploaded are bereft with necessary textual metadata that can be used in order to rank the images for a given query. Let us consider an example that is observed in the given data set.

For the above example images, we got cosine similarity measure equivalent to 34.5 percent. This is far too less. In order to improve on to, various word2vec training corpus has been experimented with. One such example is to use Glove Model. Using

Tags	Title	Description
travel india tourism agrafort agra mughal mehtabbagh	Agra fort entrance Camel with fort in background	Agra Fort, India

Table 5. Example Images from location query 'Agra Fort'

Glove Model was not straightforward as most of the textual metadata was obtained from users who aren't proficient enough in spellings, grammar etc. In the above example, "agra fort" has been interpreted as "argafort" and "mehtab bagh" has been interpreted as "interpreted" by the user. There needs further data prepossessing which needs essential text correction process which is out of scope in our thesis. Using Glove Model and removing the words stated above, the similarity rose to 72 percent.

Adding the first sentence of Wiki description of Agra Fort has a drastic improvement in cosine similarity. The Cosine similarity measure increased to 91.6 percent from 34.5 percent and Glove Model similarity measure increased to 98.52 percent from 72 percent.

In this approach, class information that is obtained from Flickr dataset is used to expand the query vector. In order to do this, various supervised machine learning algorithms are used to determine the location a given image belongs to. Using the resultant class (location) each of these images belong to, the query vector (bag of words representation of title, description and tags of the images) is strengthened with the first two sentences from the Wikipedia. Using this query vector, we get the list of images in a cosine similarity ranked order.

The training data used for training is varied, *i.e.*, varying the size of training data for every member. A very simple approach is the k-fold cross-validation where k different models are trained on k different subsets of the training data. The predictions

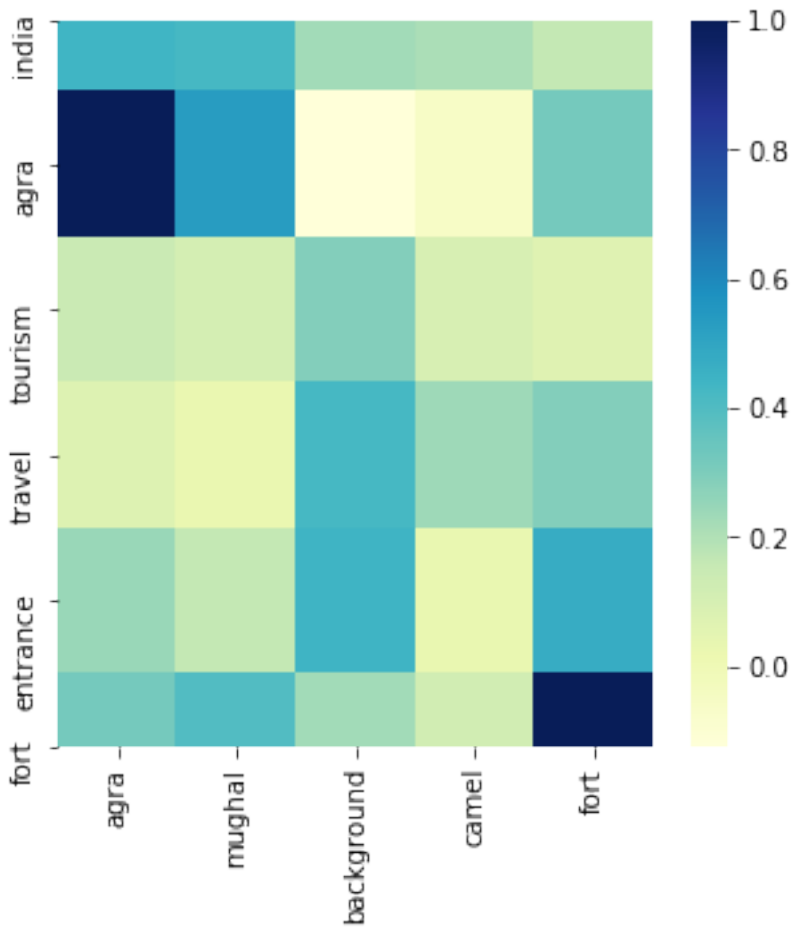

Figure 12. Heat Plot for Cosine Similarity Model without Wiki sentence

from all the models combined can be used as the final prediction. This way, a final class can be predicted for all of the images.

5.1.1.2 Class Evaluation for Input Query

In order to evaluate the class an input query belongs to, different supervised learning algorithms are used. Later, a ranking algorithm predicts the best match based on the majority voting. Below are the few machine learning algorithms that have been used for predicting the class.

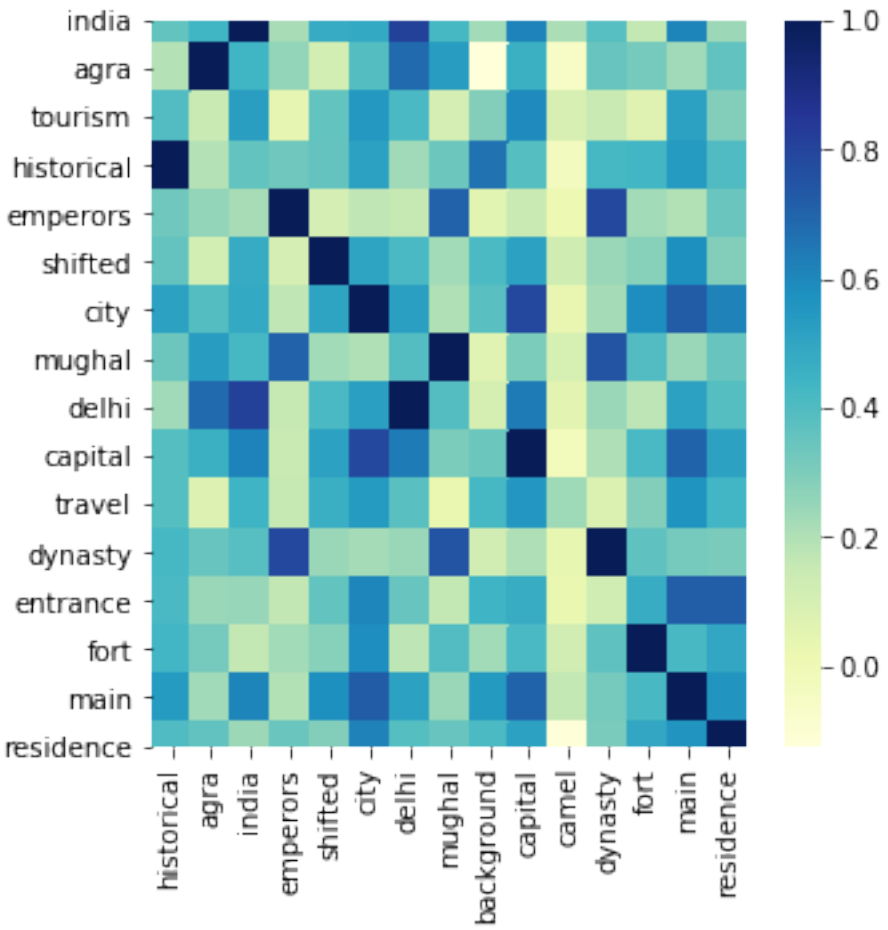

Figure 13. Heat Plot for Glove Model with Wiki sentence

- Naive Bayes
- Support Vector Machines
- Deep Learning
- Logistic Regression
- Word2vec and Logistic Regression
- Doc2vec and Logistic Regression

Finally, a majority ranking algorithm is employed inorder to determine the final results. In this we employ a count based mechanism in order to determine the final

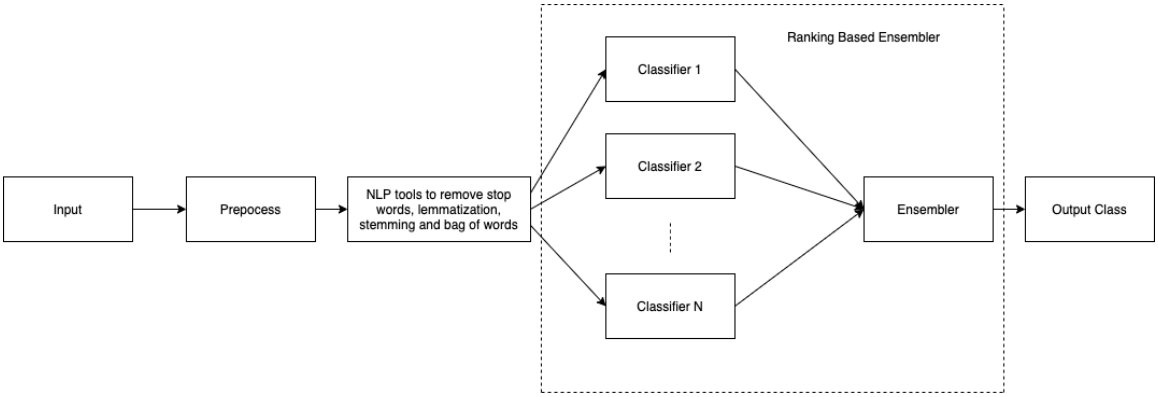

Figure 14. System for finding resulting class

class a query belongs to. Here we have taken the Count (C) equivalent to 4. If there are atleast four supervised algorithms predicting the same class, we can come to a conclusion that a query belongs to the given class (landmark determined by Flickr).

5.1.1.3 Similarity Ranking

A majority ranking algorithm is employed in order to predict the final class an image belongs. Based on the class an image belongs to, we strengthen the query by concatenating the first two sentences from Wikipedia. If half majority is not attained, the original query will be retained without any modification. It is observed that this step improves the precision rate and alters the ranking of different images for a particular query. Since our major objective is to get the diverse images that are relevant for a given query location, this method is employed although it changes the image ordering.

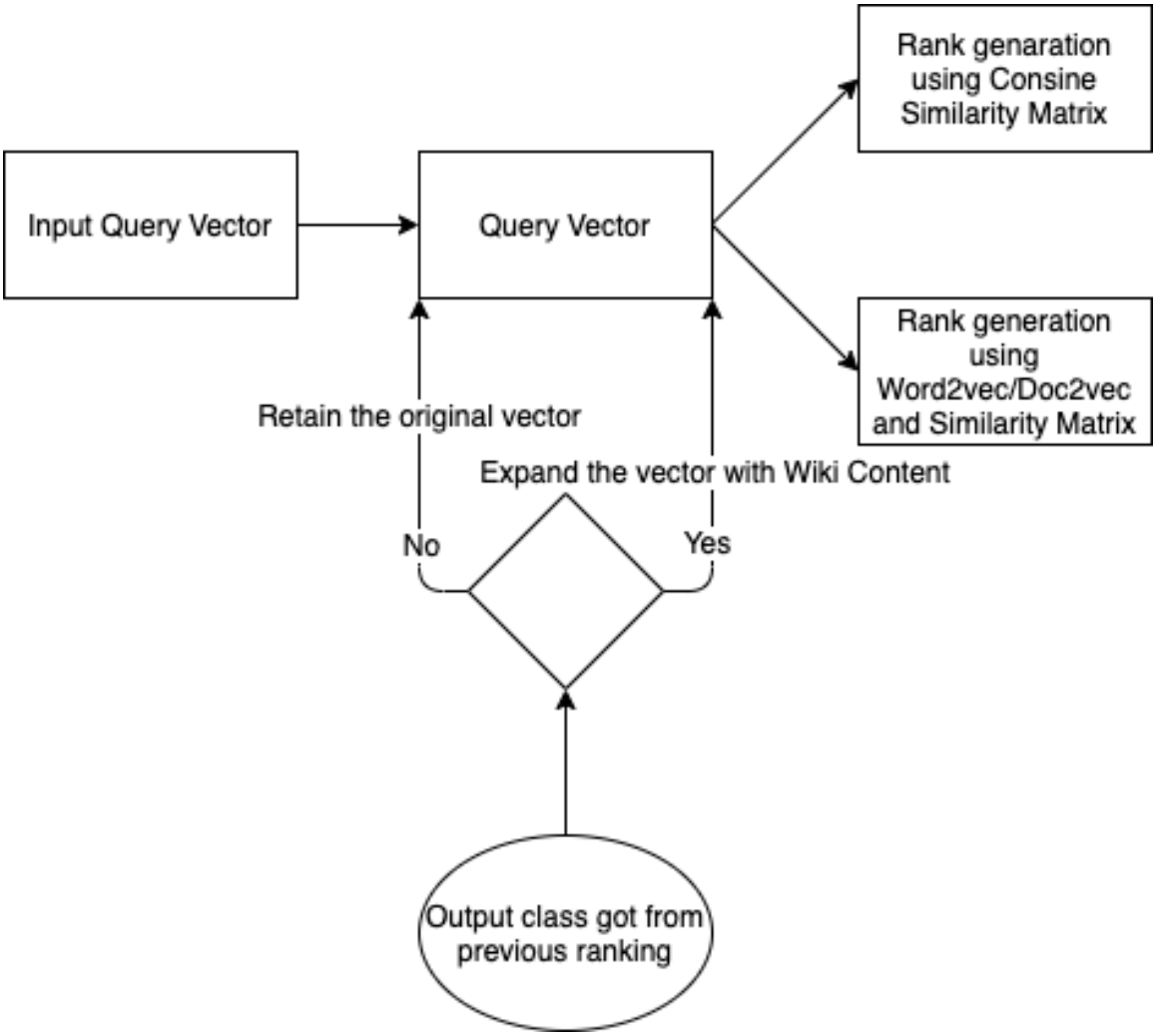

Figure 15. System for similarity ranking

ML Algorithm	Class Predicted
Naive Bayes	True
Support Vector Machines	True
Deep Learning	False
Logistic Regression	True
Word2vec and Logistic Regression	True
Doc2vec and Logistic Regression	True

Table 6. Example image for which Class indicated as Agra Fort with 5-1 voting for image 2976176. For this image, we will be strengthening the query with Wikipedia content

ML Algorithm	Class Predicted
Naive Bayes	False
Support Vector Machines	False
Deep Learning	False
Logistic Regression	False
Word2vec and Logistic Regression	True
Doc2vec and Logistic Regression	True

Table 7. Class indicated as Agra Fort with 2-4 voting for image 3259863724. The original query will be retained as we are not able to predict the right class using supervised learning

5.1.2 Relevant Search Results using Visual Metadata

Previous section explained how we are able to leverage textual meta data in order to retrieve relevant results. In this section, we will be using visual descriptor in order to retrieve the relevant results. As explained in the previous section, visual descriptors are of two types:

- Using Visual Descriptors obtained from dataset.
- Using features extracted from ResNet50

5.1.2.1 Using Visual Descriptors Obtained from Dataset

Below are some of the descriptors that are used for image ranking process.

- Global Color Naming Histogram
- Global Histogram of Oriented Gradients
- Global Color Moments on HSV Color Space
- Global Locally Binary Patterns on gray scale
- Global Color Structure Descriptor

- Global Statistics on Gray Level Run Length Matrix
- Spatial pyramid representation

Along with the above mentioned descriptors, we have also used some of the descriptors that have been extracted for each of the image. The features from various combinations of these descriptors are normalized and tested using various machine learning as described in the previous section. Later, a similarity matrix is build and is used for ranking the images based on the input query.

5.1.2.2 Using Features Extracted from Resnet50

Keras provides a set of state-of-the-art deep learning models along with pre-trained weights on ImageNet. These pre-trained models can be used for image classification, feature extraction, and transfer learning. In order to extract features from the images, ResNet50 is used to obtain the feature vectors. These feature vectors are later used to test classification and a similarity matrix is formed using them. Using this similarity matrix, we rank the given set of images for a query input.

5.2 Getting Divergent Results

Getting relevant results is only a part of the thesis. In this thesis, our objective is to get images that are divergent enough to give a well rounded picture of the query location. For this, two different algorithms are analysed. One is the modified KNN algorithms and other using various clustering algorithms are used based on similarity measures. They share the idea of creating a similarity graph (potentially complete) in which each vertex represents an image for one point of interest, and each

edge represents the similarity between two images. Different similarity metrics and different set of features are experimented with. Next, various clustering algorithms are explained along with different techniques used in order to combine them. We have experimented with various features (both visual and textual) and similarity measures (cosine similarity, euclidean similarity).

5.2.1 Introduction to Clustering Based Diversification

Naturally, establishing an order on the given set of data objects requires an understanding of the fundamental characteristics and features of the media and the use of data structures appropriate for these features. In social media, we may not have prior knowledge about the explicit features of data. This is the case, for example, when we have "black-box programs" that can compare two objects or when the similarity of the pair is simply evaluated subjectively by users. In both cases, we can obtain information about distances and/or similarities between pairs of objects, but there are no explicit features that one can use as a basis for an index structure. In these situations, we can rely on clustering techniques that do not need explicit features to operate. Before looking into various clustering algorithms, we need take into consideration some of the measures as stated below into order to ensure the cluster quality.

The quality of a clustering scheme can be quantified in various ways, some of which may conflict with each other. The appropriate quality measures are application dependent. Let C = C1,C2, . . . ,Ck be the set of clusters obtained by processing the objects in a given set, S. The following are some commonly used cluster quality measures

Cluster Diameter - The diameter of a cluster is the maximum distance (or dissimilarity) of objects included in the cluster. The problem of partitioning a given set of entities into k clusters, such that the sum of the diameters of the clusters is minimum, is known to be NP-complete for k greater than 3

Cluster Homogeneity/Compactness - One can quantify the homogeneity (or compactness) of a cluster by computing the sum or average of all similarities of object pairs in the cluster:

$$Compactness(C) = \sum_{o_k != o_l; o_k, o_l \in C_l} sim(O_k, O_l) \tag{5.1}$$

A method that is more efficient to compute, and thus often used, is the sum-ofsquares, which is the sum of squared distances of all objects in the cluster from the corresponding cluster centroid (or representative). The minimum sum-of-squares clustering problem of partitioning a given set of entities into k clusters in such a way that the sum of squared distances is minimized is known to be NP-hard [Aloise et al., 2008] in the Euclidean space.

A related quality measure is the root-mean-square-error (RMSE) measure, which is the average of the squared distances from the objects to the cluster centroid. Given a clustering scheme C = C1,C2, . . . ,Ck, the root-mean-squared error.

5.2.1.1 Input

In our clustering model, the top re-ranked results are the images that are used as our input. These are the images that we get from our previous step based on the input query - landmark or description based query. In this work we are limiting it to a maximum of 100 images as retrieving more results might be computationally

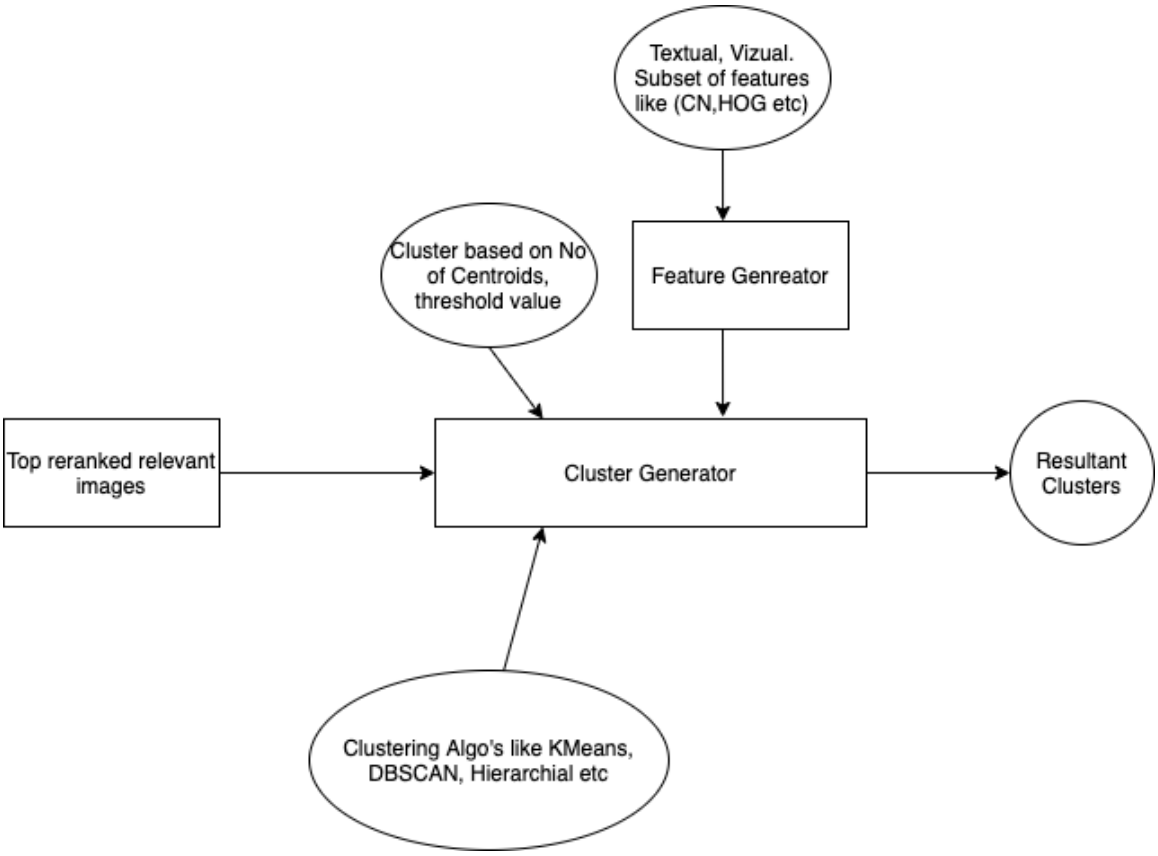

Figure 16. System for Clustering

challenging. In future, we might what it to be be around 20 percent of all the relevant results.

5.2.1.2 Cluster Variables - Number of Clusters vs Threshold

Defining a threshold is always a better approach that determining the number of clusters. The reason being that threshold always acts as a parameter that balances both the compactness and separation between the clusters. We used both of there for our experimentation. Since fusion mechanism needs the number of clusters to the

same, we have ensured that we take same number of clusters for all the clustering algorithms.

5.2.1.3 Feature Generation

Both Textual and Visual Features are explored. Visual features are the ones which are diversely studies as we looking to diverse results based on visual aspects and not the description. In order to do that, various combinations of the input features are tested and different scores like Silhouette Coefficient, Calinski-Harabasz index and Davies-Bouldin index are ranked for each of the clustering algorithms. Based on all these indexes, we are taking the top 5 resultants for each of the clustering algorithm for fusion.

5.2.2 Approach for Clustering

As explained in the background literate various algorithms are employed in getting the divergent results.

- Metis Clustering
- Spectral Clustering
- DBSCAN
- Agglomerative
- Birch
- Metis Clustering
- K Means

For most of these algorithms, we used scikit learn library inorder to create the clusters.

One algorithm we didn't any implemented on our own is Max a Min K Mean Clustering Algorithm, a modified version of K Means algorithm.

5.2.2.1 Max a Min Clustering

K-Means Clustering described above minimizes the sum of the intra-cluster variances. Its simplicity and efficiency have established it as a popular means for performing clustering across different disciplines. There is a limitation that k-Means suffers from. The final clusters that are formed are heavily dependent on the initial positions of the cluster centers. If there is any bad initialization, it can easily give us very poor results. In order to overcome this, we can use the so called Max a Min Clustering Algorithm. This algorithm although increases the time taken over K Mean algorithm, it is more fine tuned to different data samples. This way, our clusters will not be confined to the order of data samples in a dataset.

5.2.2.2 Cluster Fusion

Cluster analysis is usually employed in the initial stage of understanding a raw data, especially for new problems where prior knowledge is minimal. A particular clustering model may produce an acceptable result for one dataset, but possibly become ineffective for others. Generally, there are two major challenges inherent to clustering algorithms. First, different techniques discover different structures (e.g., cluster size and shape) from the same set of data objects (Fred and Jain 2005),(Xue, Chen, and Yang 2009). For example, k-means that is probably the best known

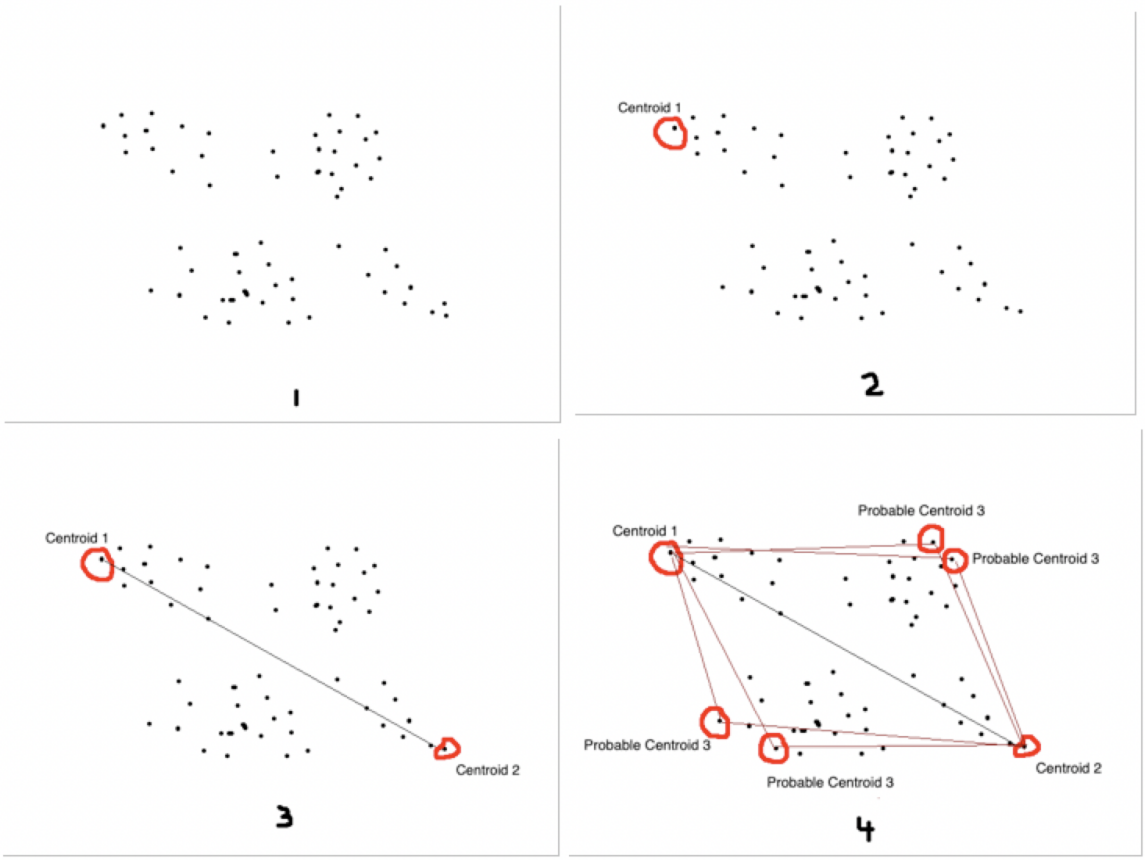

Figure 17. Selection of Cluster centers in Clustering

technique is suitable for spherical-shape clusters, while single-linkage hierarchical clustering is effective to detect connected patterns. This is due to the fact that each individual algorithm is designed to optimize a specific criterion. Second, a single clustering algorithm with different parameter settings can also reveal various structures on the same dataset. A specific setting may be good for a few, but not all datasets. Users encounter these challenges, which consequently make the selection of a proper clustering technique very difficult. The solution to this dilemma is cluster fusion. Cluster fusion can be done based on Homogeneous fusion approaches or Heterogeneous fusion approaches. Homogeneous approaches are based on different number of clusters, data sampling, feature selection etc. Heterogeneous fusion is to include different

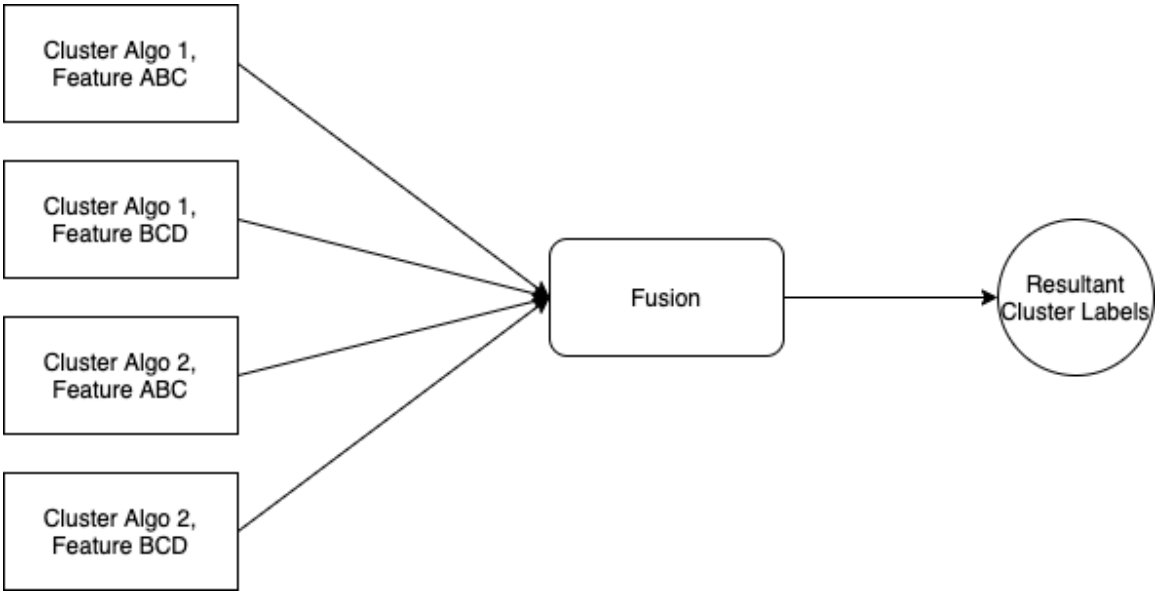

Figure 18. Cluster Fusion

clustering algorithms. In our work, we will be choosing one from Homogeneous approach *i.e.*, feature selection (combinations of features) and different clustering algorithms from Heterogeneous techniques. We call this the mixed heuristics.

The major problem with fusion in unsupervised learning is that the initial labels are not known. Two different techniques are discussed here.

5.2.2.2.1 Voting Based Methods

One of the algorithm implemented in our work is the Hard Correspondence voting is the technique of relabelling and voting. Find the correspondence between the labels in the partitions and fuse the clusters with the same labels by voting [DuFr03,DWH01]. The major advantages of using this method is that it minimize match costs and match is made to a reference clustering or match in a pairwise manner. One of the main problem with using this technique is that in most cases, clusters do not have one-to-one correspondence

In our thesis, since we are clustering only around 100 data points, we are able

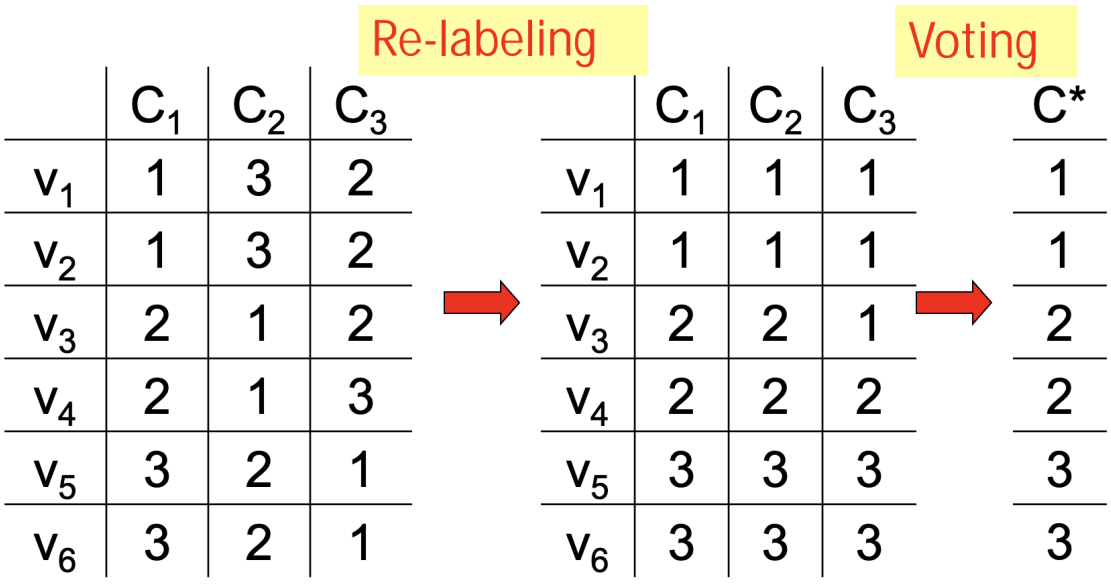

Figure 19. Voting Based Fusion Source: Jing Gao - University of Buffalo course on Data Mining and Bioinformatics

to find a hard correspondence for all of the queries that are tested. A recursive O(N*N) algorithm has been implemented in our thesis work. Although we are able to get good results using this algorithm, there might be few datasets where a Hard Correspondence can run into an infinite loop. Other voting based methods like Incremental Voting - a model is initially developed in the studies of (Dimitriadou, Weingessel, and Hornik 2001), (Frossyniotis, Pertselakis, and Stafylopatis 2002) and later generalized by (Ayad and Kamel 2007) has been used in our work which is more stable than our own algorithm. We used the OpenEnsembles package (https://naeglelab.github.io/OpenEnsembles/) for implementing these fusion model.

5.2.2.2.2 Graph-based Approach

This family of algorithms makes use of the graph representation to solve the cluster ensemble problem (Domeniconi and Al-Razgan 2009),(Fern and Brodley 2004). In this approach, a weighted graph is first constructed from the clustering ensemble. Then, the graph is partitioned in to K parts to produce the final clustering using any graph partitioning techniques. As discussed in the background literature, Graph-based Consensus Clustering, Cluster-based Similarity Partitioning Algorithm are example approaches used in our thesis. We used the OpenEnsembles package (https://naeglelab.github.io/OpenEnsembles/) for implementing these fusion model.

5.2.2.3 Selecting Images from the Clusters

5.2.2.3.1 Selection Based on Cluster Size

The clusters are ordered(non increasing) based on the size of each of the clusters. Ratio of number of images in the cluster to the total number of images that are in all the clusters is calculated. This ratio is then multiplied with the total number of image that needs to be retrieved (10,20,30). Based on this score, the images are selected in a top down approach(non increasing order). There is a chance that the smallest cluster might be left out in this approach as it can include outliers.

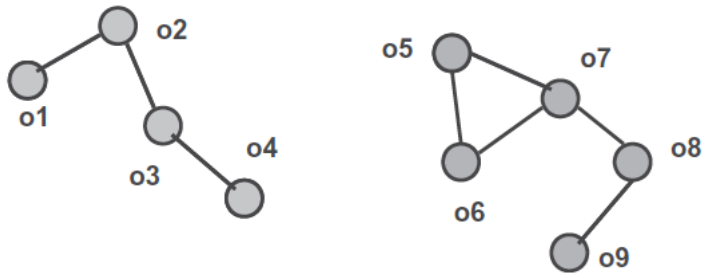

Figure 20. Connected Component Graph Based Diversification

5.2.2.3.2 Sequential Selection

The clusters are ordered based on the size(non increasing) of each of the clusters. The images are then picked starting from the largest to the lowest sized cluster sequentially. None of the clusters are left out in this process.

5.2.3 Other Approaches for Diversification

Diversification based on clustering is the major procedure followed in our diversification algorithm. There are two other approaches that are used in our work as a baseline models but not extensively tested taking run time and final results into account. For both of these methods be created a sparse graph with K nearest neighbours connected to each of the node to the K nearest neighbours. In order to keep the graph spares and improve on the runtime we have kept K to be 5.

5.2.3.1 Connected Components

The connected components scheme works on the premise that if two objects are related, there must be some (direct or indirect) linkage between these two objects; otherwise, if two objects do not have any path between them, then they must be different. Based on this premise this method searches for the groups of nodes that are pairwise reachable and labels each group as a same/different. A major advantage of connected components is the efficiency of the process: starting from an arbitrary node, one can follow all the adjacent edges until no more adjacent edges can be found; the process then can be repeated starting from any of the remaining nodes until no unvisited nodes/edges remain. The cost of the process is O(|V| + |E|), that is, linear in the graph size. A major disadvantage of this approach, on the other hand, is that the resulting object that are aught to be similar may contain objects that are quite dissimilar from each other. For example, objects o1 and o4 in Figure 8.2 are in the same cluster, although they are not neighbors. Furthermore, they are not even in each other's two-edge neighborhood. When connected components contain such long chains, the objects at the remote ends of these chains may be very different from each other.

5.2.3.2 Diversifying K Nearest Neighbours

This is the first method used in our thesis that is later deprecated due to the time complexity in this approach. In this approach after we get the relevant results we form a KNN graph out of it connecting each of the components. Now using the technique discussed in Clustering to find the centoids, we extend it to the entire 100

nodes. In order to understand it's complexity, let us compare it with the sub problem of finding centroids in Clustering. Here is an example where we have taken the K value as 5. Let is consider the time taken for this algorithm to be t. In our KNN algorithm, we have around 100 nodes (this can increase based on size of the dataset). Since the time complexity of the algorithm is N*N, we can say that 100*100/5*5 = 400t. Comparatively, it is 1:400.

Chapter 6

EXPERIMENTS AND RESULTS

In this section, we define and explain the evaluation metrics of this content detection task. We analyze both speed and accuracy in our experiments. Also, the results are measured on both the test data and the credibility data. The aggregated data has around 40 different location with 300 images which is used for our evaluation. Also, for each location, photos were manually annotated for relevance and diversity. Ground truth was generated by a small group of expert annotators with advanced knowledge of location characteristics. Software tools were specifically designed to facilitate the annotation process. In our thesis, we will be dividing this section into 2 parts, analysing results of relevant and divergent results separately. Separate runs will be run to evaluate both the test set data and credibility data. There runs will initially make comparisons among different models discussed in our thesis and the best among then will be compared over state of art models.

One measure we will be using throughout our evaluation process is precision and recall. Precision is defined as the proportion of all predictions which are from the positive class *i.e.*, is calculated by dividing the number of true positives with the total number of true positives and false positives. The recall value is calculated by dividing the number of true positives by the number of true positives and false negatives. Using the credibility data, we evaluate it on the top 10, 20 and 30 image results.

$$Precision(P) = \frac{truepositives}{truepositives + falsepositives} \tag{6.1}$$

$$Recall(R) = \frac{truepositives}{truepositives + falsenegatives} \tag{6.2}$$

6.1 Relevant Results

In this section we will look into the results obtained using both textual metadata and visual data. Before looking into our similarity measure, let us look into the supervised learning results that we use to identify the class and strengthen our query using Wiki data. Initially we have taken only description given for each of the image into consideration as our input. This is because, there are some of the images which didn't have any title or tags. This didn't give us a good results as shown in Table 8 .

From the above results you can observe that we are able to get the best predictions using SVM and Logistic Regression over using Word2Vec, Dec2Vec and Neural Networks. The reason can be attributed to the fact that the data sample for each class is on average of 293 without any filters and of 257 using all the prefilters. One supersizing result we found is that using word training corpus like Word2vec, Doc2vec etc. performed worse than the simple Logistic Regression and SVM. The reason can be attributed to the fact that tags and titles are the ones which are majorly contributing for classification (you can take a look into the table results above).

‘

Let's take a look into Visual Descriptors. In our experimentation we have used all the combinations to get the best feature set. 'CM3x3','CM','CN', 'CN3x3', 'CSD', 'GLRLM', 'GLRLM3x3', 'HOG', 'LBP', 'LBP3x3' are the visual descriptor feature set that are used. Since we have around 10*11/2 = 55 combinations and different algorithms are analyzed(55*5 different ML algorithms), only the top 20 and bottom

Figure 21. Input Image for query 'Altes Museum'

Figure 22. Nearest images to the input image belonging to 'Altes Museum'. From left to right and top to bottom belonging to the landmarks - castillo de san marcos, acropolis athens,castillo de san marcos, castillo de san marcos

Figure 23. Example 2: Sculpture from Atlas Museum

Figure 24. Images retrieved using Visual Metrics. 3 among top 6 similar images belonging landmarks other than Atlas Museum

Accuracy results based on textual descriptors (only image description)		
Filter Used	Learning Algorithm	Accuracy Measure
None	Navie Bayes	0.54
Face Detection	Navie Bayes	0.51
Face and Blur	Navie Bayes	0.57
None	SVM	0.55
Face Detection	SVM	0.52
Face and Blur	SVM	0.54
None	LogisticRegression	0.56
Face Detection	LogisticRegression	0.56
Face and Blur	LogisticRegression	0.55
None	Word2vec and Logistic Regression	0.56
Face Detection	Word2vec and Logistic Regression	0.52
Face and Blur	Word2vec and Logistic Regression	0.53
None	Doc2vec and Logistic Regression	0.50
Face Detection	Doc2vec and Logistic Regression	0.45
Face and Blur	Doc2vec and Logistic Regression	0.51
None	BOW with Keras	0.57
Face Detection	BOW with Keras	0.53
Face and Blur	BOW with Kerass	0.51

Table 8. Table showing accuracy measures taken description of the images

20 combinations have been reported. The accuracy measure was less than that what we get using textual meta data. These visual metrics will be useful because not all the images taken from flickr dataset will have title and tags attached. Ideally we assume that a user might not be providing us any title, tag or description for that image. In these cases, visual descriptor data might be useful. Ideally, there can be a way where textual metadata and visual data can both be used inorder to evaluate a class the given image belongs to. This work hasn't been included in our thesis.

Accuracy results based on textual descriptors		
Filter Used	Learning Algorithm	Accuracy Measure
None	Navie Bayes	0.89
Face Detection	Navie Bayes	0.88
Face and Blur	Navie Bayes	0.89
None	SVM	0.93
Face Detection	SVM	0.94
Face and Blur	SVM	0.94
None	LogisticRegression	0.94
Face Detection	LogisticRegression	0.95
Face and Blur	LogisticRegression	0.94
None	Word2vec and Logistic Regression	0.84
Face Detection	Word2vec and Logistic Regression	0.86
Face and Blur	Word2vec and Logistic Regression	0.83
None	Doc2vec and Logistic Regression	0.86
Face Detection	Doc2vec and Logistic Regression	0.85
Face and Blur	Doc2vec and Logistic Regression	0.87
None	BOW with Keras	0.92
Face Detection	BOW with Keras	0.93
Face and Blur	BOW with Kerass	0.91

Table 9. Table showing accuracy measures taken description, tags and title of the images

Finding the best fit, for each of the landmark location, 100 images that are relevant to the query have been extracted. Below are the accuracy measure for each of the landmark location:

For the results we observed from Table 12 and Table 13, we can see that the results haven't improved drastically on using Glob Vector of Words compared to using Cosine Similarity matrix. But one observation we can see from the results is that

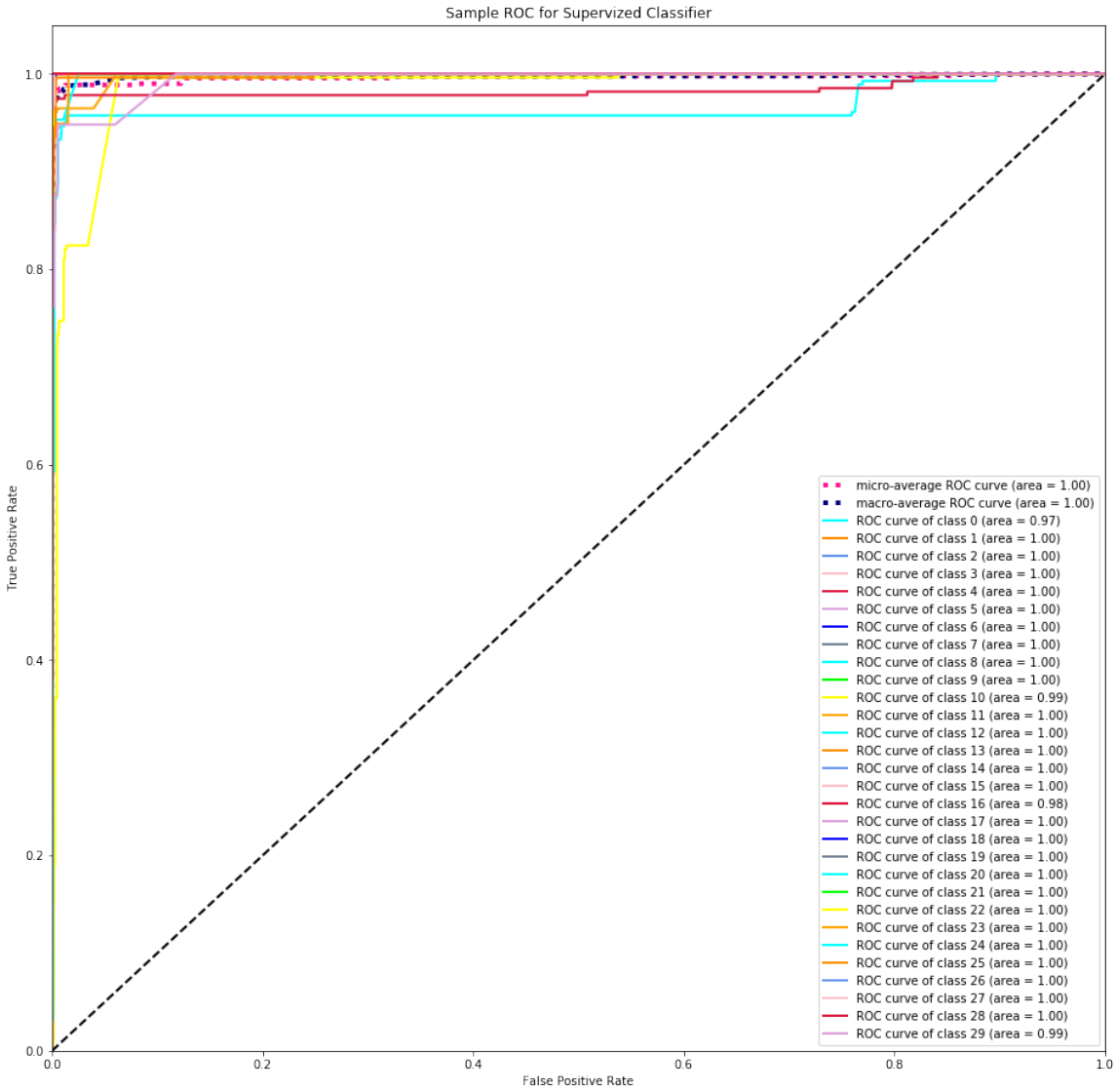

Figure 25. ROC Curve for all the 30 landmarks using LogisticRegression

Top 20 accuracy results based on visual descriptors		
Supervised Algorithm	Visual Features Used	Accuracy Measure
CNN	ResNet50	0.5704123686963483
LinearSVC	CM3x3 CM CN CN3x3 CSD HOG LBP LBP3x3	0.4673141576391483
LinearSVC	CM3x3 CM CN CN3x3 HOG LBP LBP3x3	0.4650728427344042
LinearSVC	CM3x3 CM CN CN3x3 CSD HOG LBP	0.44676877101232726
LinearSVC	CM3x3 CM CN CN3x3 HOG LBP3x3	0.44676877101232726
LinearSVC	CM3x3 CM CN HOG LBP LBP3x3	0.44676877101232726
LinearSVC	CM CN CN3x3 CSD HOG LBP LBP3x3	0.4449010085917071
LinearSVC	CM CN CN3x3 HOG LBP LBP3x3	0.44378035113933506
LinearSVC	CM3x3 CM CN CN3x3 CSD HOG LBP3x3	0.44378035113933506
LinearSVC	CM3x3 CM CN CSD HOG LBP LBP3x3	0.442659693686963
LinearSVC	CM3x3 CN CN3x3 HOG LBP LBP3x3	0.44153903623459095
LinearSVC	CM3x3 CM CN3x3 HOG LBP LBP3x3	0.44116548375046694
LinearSVC	CM3x3 CN CN3x3 CSD HOG LBP LBP3x3	0.4404183787822189
LinearSVC	CM3x3 CM CN3x3 CSD HOG LBP LBP3x3	0.4400448262980949
LinearSVC	CM3x3 CM CN CN3x3 HOG LBP	0.4385506163615988
LinearSVC	CM3x3 CM CN CSD HOG LBP3x3	0.42771759432200224
Naive-Bayes	CM3x3 CM CN CSD HOG LBP LBP3x3	0.423234964512514
LinearSVC	CM3x3 CM CN CSD HOG LBP	0.4209936496077699
Naive-Bayes	CM3x3 CM CN CN3x3 CSD HOG LBP LBP3x3	0.4206200971236459
LinearSVC	CM CN CN3x3 CSD HOG LBP	0.4202465446395219

Table 10. Top 20 accuracies based on visual descriptors

Bottom 20 accuracy results based on visual descriptors		
Supervised Algorithm	Visual Features Used	Accuracy Measure
LinearSVC	CM CN3x3 CSD GLRLM LBP LBP3x3	0.06238326484871124
LinearSVC	CN GLRLM LBP	0.06238326484871124
LinearSVC	CM CN3x3 GLRLM LBP LBP3x3	0.06238326484871124
LinearSVC	CM CN CN3x3 GLRLM LBP LBP3x3	0.06462457975345536
LinearSVC	CM CN CN3x3 CSD GLRLM LBP LBP3x3	0.06462457975345536
LinearSVC	CSD GLRLM HOG LBP3x3	0.0653716847217034
LinearSVC	GLRLM HOG LBP3x3	0.0653716847217034
LinearSVC	CN3x3 CSD GLRLM LBP LBP3x3	0.06611878968995144
LinearSVC	CM3x3 CM CN CN3x3 CSD GLRLM HOG LBP LBP3x3	0.06611878968995144
LinearSVC	CM3x3 CM CN CN3x3 GLRLM HOG LBP LBP3x3	0.06611878968995144
LinearSVC	CM3x3 CM CN3x3 GLRLM HOG	0.06649234217407546
LinearSVC	CM3x3 CN CN3x3 CSD GLRLM HOG LBP3x3	0.06649234217407546
LinearSVC	CM3x3 CN CN3x3 GLRLM HOG LBP3x3	0.06649234217407546
LinearSVC	CN CSD GLRLM LBP	0.06686589465819948
LinearSVC	CM CN CSD GLRLM LBP3x3	0.06686589465819948
LinearSVC	CM CN GLRLM LBP3x3	0.06686589465819948
LinearSVC	CM3x3 CM CN3x3 GLRLM LBP LBP3x3	0.06761299962644751
LinearSVC	CM3x3 CM CN3x3 CSD GLRLM LBP LBP3x3	0.06761299962644751
LinearSVC	CM CN3x3 GLRLM HOG LBP LBP3x3	0.06798655211057153
LinearSVC	CM CN3x3 CSD GLRLM HOG LBP LBP3x3	0.06798655211057153
LinearSVC	CM3x3 CM CN3x3 CSD GLRLM LBP	0.06836010459469556
LinearSVC	CM3x3 CN3x3 CSD GLRLM LBP	0.06836010459469556

Table 11. Bottom 20 accuracies based on visual descriptors

the accuracies improved for locations which are closer and similar. Example is Altes Museum and Neues Museum. Both of there museum are located in Berlin, Germany.

Table 12. Accuracy for top 100 results for each landmark query using cosine similarity

Landmark	Accuracy
Acropolis Athens	100
Agra Fort	100
Albert Memorial	98
Altes Museum	62
Amiens Cathedral	79
Angel of the North	98
Angkor Wat	84
Ara Pacis	100
Arc De Triomphe	83
Aztec Ruins	79
Berlin Cathedral	71
Big Ben	100
Bok Tower Gardens	97
Brandenburg Gate	97
Cabrillo	62
Casa Batllo	77
Casa Rosada	90
Castillo De San Marcos	99
Chartres Cathedral	99
Chichen Itza	100
Christ The Redeemer Rio	98
Civic Center SF	86
CN Tower	100
Cologne Cathedral	85
Colosseum	91
Hearst Castle	73
La Madeleine	83
Montezuma Castle	100
Neues Museum	74
Pont Alexandre iii	100

In the Figure 22, showing the top results for Altes Museum, some of the results have been recorded from different other landmarks which are located in Germany (Neues Museum, Berlin Cathedral). In the figure, there are 4 images (7195597596, 7195592888, 7195577516, 4472434082) that doesn't belong to Altes Museum. A get

Figure 26. Top results for landmark - Acropolis Athens

Figure 27. Top results for landmark - Altes Museum

Table 13. Accuracy for top 100 results for each landmark query using glove model to learn the word embeddings and cosine similarity

Landmark	Accuracy
Acropolis Athens	97
Agra Fort	96
Albert Memorial	99
Altes Museum	78
Amiens Cathedral	82
Angel of the North	92
Angkor Wat	100
Ara Pacis	100
Arc De Triomphe	100
Aztec Ruins	84
Berlin Cathedral	89
Big Ben	68
Bok Tower Gardens	100
Brandenburg Gate	100
Cabrillo	98
Casa Batllo	88
Casa Rosada	78
Castillo De San Marcos	81
Chartres Cathedral	77
Chichen Itza	100
Christ The Redeemer Rio	94
Civic Center SF	98
CN Tower	87
Cologne Cathedral	99
Colosseum	48
Hearst Castle	79
La Madeleine	89
Montezuma Castle	83
Neues Museum	82
Pont Alexandre iii	100

around to it is to leave out textual data with location values (like berlin, Germany) in this case to improve the results. Although, this helped in getting better results between the landmarks in Germany, there is a decrease in overall accuracy.

6.2 Divergent Images

In this method various clustering techniques will be evaluated. There different models are taken into consideration inorder to evaluate the clustering models. There are 3 different evaluation metrics for to evalaute metrics for which ground truth labels are not known. They are Silhouette Coefficient, Calinski-Harabasz index and Davies-Bouldin index.

6.2.1 Evaluation of Metrics for Clusters

6.2.1.1 Silhouette Coefficient

If the ground truth labels are not known, evaluation must be performed using the model itself. The Silhouette Coefficient is an example of such an evaluation, where a higher Silhouette Coefficient score relates to a model with better defined clusters. The Silhouette Coefficient is defined for each sample and is composed of two scores:

- The mean distance between a sample and all other points in the same class.
- The mean distance between a sample and all other points in the next nearest cluster.

The Silhouette Coefficient s for a single sample is then given as:

$$s = \frac{b - a}{max(a, b)} \tag{6.3}$$

6.2.1.2 Calinski-Harabasz Index

If the ground truth labels are not known, the Calinski-Harabasz index - also known as the Variance Ratio Criterion - can be used to evaluate the model, where a higher Calinski-Harabasz score relates to a model with better defined clusters.

The index is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters (where dispersion is defined as the sum of distances squared). For a set of data of size which has been clustered into clusters, the Calinski-Harabasz score is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion:

$$s = \frac{t_r(B_k)}{t_r(W_k)} \times \frac{n_E - k}{k - 1} \tag{6.4}$$

where is trace of the between group dispersion matrix and is the trace of the within-cluster dispersion matrix

6.2.1.3 Davies-Bouldin Index

If the ground truth labels are not known, the Davies-Bouldin index can be used to evaluate the model, where a lower Davies-Bouldin index relates to a model with better separation between the clusters.

This index signifies the average 'similarity' between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves. Zero is the lowest possible score. Values closer to zero indicate a better partition.

The index is defined as the average similarity between each cluster for and its

most similar one . In the context of this index, similarity is defined as a measure that trades off:

- Si , the average distance between each point of cluster and the centroid of that cluster – also know as cluster diameter.
- Dij the distance between cluster centroids i and j.

Best performances in terms of Silhotte (for all cut off points N = 5; 10; 20; 30; 40; 50) were obtained by using the following visual descriptors: global color naming histogram, histogram of oriented gradients 2x2, dense SIFT, locally binary pattern with uniform patterns, and global color structure descriptor. Thus, for the rest of the experiments, we used these visual descriptors and the tuned parameters. Table 1 lists all 22 descriptors which have been tested, where the bold ones are the selected descriptors.

6.2.2 Feature Evaluation Results for Various Clustering Algorithms

Like in the previous module, we have tried various combinations inorder to evaluate the results. For divergence visual descriptor data is used over textual based on observations. Although evaluation metrics performed equally well for both textual and visual data, observing the clusters manually proved that visual data is a better fit than the textual data. Also, the problem statement tries to get diverse results based on the visual forms and not the textual. Below we have stated the results of evaluating a single scan clustering algorithm, a graph based algorithm and a hierarchical algorithm. Finally we evaluate the scores based on the cluster fusion method.

Table 14 shown are the feature combinations that retrieved the top Silhouette score for K Means Clustering. The visual feature set - “CM,CN and GLRLM” attained

the top score with 0.298 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, “CN” obtained the highest score.

Top 5 scores based on Silhouette Score		
Number of Clusters	Visual Features Used	Silhouette Score
5	’CM’, ’CN’, ’GLRLM’	0.2989434739211579
10	’CN’	0.29783722205381735
5	’CN3x3’, ’CSD’, ’GLRLM’, ’LBP’	0.29680392227793156
10	’CN’, ’CN3x3’, ’GLRLM’, ’LBP’	0.29664058303418395
10	’CN’, ’GLRLM’	0.2966307414026646

Table 14. Silhouette Score for K Means Clustering

Table 15 shown are the feature combinations that retrieved the top Calinski Harabasz score for K Means Clustering. The visual feature set - “GLRLM” attained the top score with 45.24 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, “GLRLM” again obtained the highest score.

Top 5 scores based on Calinski Harabasz Score		
Number of Clusters	Visual Features Used	Calinski Harabasz Score
5	’GLRLM’	45.24740304966374
5	’CN’, ’GLRLM’	42.95820436452776
5	’GLRLM’, ’LBP’	40.9070061000691
10	’GLRLM’	40.14547291635069
5	’CM’, ’GLRLM’	39.416374951057165

Table 15. Calinski Harabasz Score for K Means Clustering

Table 16 shown are the feature combinations that retrieved the top DB score for K Means Clustering. The visual feature set - “CM’, ’CN’, ’CSD’, ’GLRLM” attained the top score with 0.99 score. All the top 5 results arr attained with 10 clusters are taken into consideration.

Top 5 scores based on DB Score		
Number of Clusters	Visual Features Used	DB Score
10	'CM', 'CN', 'CSD', 'GLRLM'	0.9968595521944511
10	'CM', 'CN', 'CSD', 'GLRLM', 'LBP3x3'	1.0152335269357469
10	'CN', 'CN3x3', 'GLRLM', 'LBP'	1.0237140887851002
10	'CM', 'CN3x3', 'CSD', 'GLRLM'	1.0363300229917878
10	'CM', 'CN3x3', 'CSD', 'GLRLM', 'LBP3x3'	1.0392660785589682

Table 16. DB Score for K Means Clustering

6.2.2.1 K Means Clustering Using Max a Min Algorithm

Table 17 shown are the feature combinations that retrieved the top Silhouette score for Max a Min K Means Clustering. The visual feature set - "CN3x3', 'CSD', 'GLRLM', 'LBP" attained the top score with 0.342 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, "CN', 'GLRLM" obtained the highest score.

Top 5 scores based on Silhouette Score		
Number of Clusters	Visual Features Used	Silhouette Score
5	'CN3x3', 'CSD', 'GLRLM', 'LBP'	0.34210792227405830
10	'CN', 'GLRLM'	0.3146302411086143
10	'CN'	0.30434129185381442
5	'CM', 'CN', 'GLRLM'	0.2981024204888229
5	'GLRLM'	0.2977424214888217

Table 17. Silhouette Score for Max a Min K Means Clustering

Table 18 shown are the feature combinations that retrieved the top Silhouette score for Max a Min K Means Clustering. The visual feature set - "GLRLM', 'LBP" attained the top score with 45.9 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, "GLRLM" obtained the highest score.

Table 19 shown are the feature combinations that retrieved the top DB score for

Top 5 scores based on Calinski Harabasz Score		
Number of Clusters	Visual Features Used	Calinski Harabasz Score
5	'GLRLM'	45.90700291635012
5	'CN', 'GLRLM'	41.41639582125482
5	'GLRLM', 'LBP'	40.9070061000691
5	'CM', 'GLRLM'	40.447891236664125
10	'GLRLM'	40.01547272365061

Table 18. Calinski Harabasz Score for Max a Min K Means Clustering

Max a Min K Means Clustering. The visual feature set - "CM', 'CN', 'CSD', 'GLRLM', 'LBP3x" attained the top score with 1.69 score. All top 5 score3s has 10 clusters.

Top 5 scores based on DB Score		
Number of Clusters	Visual Features Used	DB Score
10	'CM', 'CN', 'CSD', 'GLRLM', 'LBP3x3'	1.6935746901523352
10	'CM', 'CN', 'CSD', 'GLRLM'	1.1991445116859552
10	'CM', 'CN3x3', 'CSD', 'GLRLM'	1.0991787803633002
10	'CM', 'CN3x3', 'CSD', 'GLRLM', 'LBP3x3'	1.0558968203926607
10	'CN', 'CN3x3', 'GLRLM', 'LBP'	1.0185100202371408

Table 19. DB Score for Max a Min K Means Clustering

6.2.2.2 Spectral Clustering

Table 20 shown are the feature combinations that retrieved the top Silhouette score for Spectral Clustering. The visual feature set - "GLRLM" attained the top score with 0.308 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, "GLRLM" obtained the highest score.

Table 21 shown are the feature combinations that retrieved the top Silhouette

Top 5 scores based on Silhouette Score		
Number of Clusters	Visual Features Used	Silhouette Score
5	'GLRLM'	0.30863094447216566
10	'GLRLM'	0.3030437496115049
10	'CN', 'CN3x3', 'GLRLM', 'LBP'	0.296069208980549
5	'CN', 'CN3x3', 'CSD', 'GLRLM', 'LBP'	0.2921978294946358
5	'CN3x3', 'CSD', 'GLRLM'	0.2920096543031767

Table 20. Silhouette Score for Spectral Clustering

score for Spectral Clustering. The visual feature set - "GLRLM" attained the top score with 43.6 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, "CN,GLRLM" obtained the highest score.

Top 5 scores based on Calinski Harabasz Score		
Number of Clusters	Visual Features Used	Calinski Score
5	'GLRLM'	43.59484513659807
5	'CN', 'GLRLM'	40.54937851492339
5	'GLRLM', 'LBP'	39.15792884115711
10	'CN', 'GLRLM3x3'	38.37290360532748
10	'GLRLM3x3'	38.08235552186784

Table 21. Calinski Harabasz Score for Spectral Clustering

Table 22 shown are the feature combinations that retrieved the top Silhouette score for Spectral Clustering. The visual feature set - "CN3x3', 'CSD', 'GLRLM3x3" attained the top score with 1.02 score. All top 5 are from 10 cluster setting.

6.2.2.3 BIRCH Hierarchical Clustering

Table 23 shown are the feature combinations that retrieved the top Silhouette score for BIRCH Hierarchical Clustering. The visual feature set - "CM, CN, GLRLM" attained the top score with 0.26 score. For this we have consider the number of

Top 5 scores based on DB Score		
Number of Clusters	Visual Features Used	DB Score
10	'CN3x3', 'CSD', 'GLRLM3x3'	1.029493579008237
10	'CN', 'CN3x3', 'GLRLM', 'LBP3x3'	1.0322163215330526
10	'CN', 'CSD'	1.035094240956941
10	'CM', 'CN3x3', 'CSD', 'GLRLM3x3'	1.0381269834687752
10	'CM', 'CN', 'CN3x3', 'CSD', 'GLRLM3x3'	1.0430403235578292

Table 22. DB Score for Spectral Clustering

clusters as 5. When 10 clusters are taken into consideration, "CN, GLRLM" obtained the highest score.

Top 5 scores based on Silhouette Score		
Number of Clusters	Visual Features Used	Silhouette Score
5	'CM', 'CN', 'GLRLM'	0.2689434739211579
5	'CM'	0.25220578317357238
10	'CN', 'GLRLM'	0.2460266304676414
5	'CN3x3', 'GLRLM', 'LBP'	0.23860331569227927
5	'CN', 'CN3x3', 'GLRLM', 'LBP'	0.23450589618393034

Table 23. Silhouette Score for BIRCH Hierarchical Clustering

Table 24 shown are the feature combinations that retrieved the top Calinski Harabasz score for BIRCH Hierarchical Clustering. The visual feature set - "CN, GLRLM" attained the top score with 43.03 score. For this we have consider the number of clusters as 5. When 10 clusters are taken into consideration, " GLRLM" obtained the highest score.

Table 25 shown are the feature combinations that retrieved the top DB score for BIRCH Hierarchical Clustering. The visual feature set - "CM', 'CN', 'CSD', 'GLRLM" attained the top score with 0.99 score. All top scores are observed for 10 clusters.

Top 5 scores based on Calinski Harabasz Score		
Number of Clusters	Visual Features Used	Silhouette Score
10	'GLRLM'	43.03049663742474
5	'CN', 'GLRLM'	42.36452776958204
5	'CM', 'GLRLM'	41.105716541637495
5	'GLRLM', 'LBP'	40.6100069190700
5	'GLRLM'	40.16350691454729

Table 24. Calinski Harabasz Score for BIRCH Hierarchical Clustering

Top 5 scores based on DB Score		
Number of Clusters	Visual Features Used	DB Score
10	'CM', 'CN', 'CSD', 'GLRLM'	0.9944511968595521
10	'CM', 'CN', 'CSD', 'GLRLM', 'LBP3x3'	1.0157469015233526
10	'CN', 'CN3x3', 'GLRLM', 'LBP'	1.0202371408878510
10	'CM', 'CN3x3', 'CSD', 'GLRLM'	1.0329917878633002
10	'CM', 'CN3x3', 'CSD', 'GLRLM', 'LBP3x3'	1.0398968226607855

Table 25. DB Score for BIRCH Hierarchical Clustering

6.3 Different Runs for Landmark Query Without Pre-filtering

In this section different runs with different combinations are submitted. Here, each run will depend on what method is previously followed. If we have checked which class a query belongs to *i.e.*, used supervised algorithms and voting based mechanism to determine the class, then it is stated as "Supervised". Else, if we only rank all the vectors based on the cosine similarity, then we call it as "None". Image Diversification might be "Null" *i.e.*, we are just choosing from the top relevant results or use clustering methods like K Means, Spectral etc. If Clustering approach, we are choosing the number of clusters between 5 and 10. Finally, visual features are selected based on the results that are obtained from the previous steps. For others (that are not following clustering for diversification) we are using all the visual features(reducing them by doing a PCA).

In the table 26, 12 different runs and their settings are described. We are not using any pre-filtering technique (Face Detection, Image Blur etc.) in these runs.

Different Runs and their Settings			
Run	Image Diversification	Number of Clusters	Diversification Features
1	None	NA	All
2	None	NA	All
3	K Means	5	CM,CN,GLRLM
4	K Means	10	CN
5	Spectral	5	GLRLM
6	Spectral	10	GLRLM
7	Fusion - Voting	5	NA
8	Fusion - Voting	10	NA
9	Fusion - Graph	5	NA
10	Fusion - Graph	10	NA

Table 26. Table showing settings of different runs

Table 27, shows the Precision, Recall and F1 Scores for development and test sets. As to reiterate, here recall scores are calculated in a peculiar way. The team that developed the dataset for this MediaEval Challenge, has 20 different clusters that are unknown to the participants. Now the top images that are selected using our trained model are matched with their clusters, *i.e.*, they check how many clusters of theirs are filled using our images. For example, if 20 images of your match the 20 clusters of theirs *i.e.*, each image of yours belongs to a different cluster of theirs, your recall rate (R) would be equivalent to 1. For this reason, we are calling recall here as cluster recall(CR).

On the development data set. the cluster recall rate is high on Spectral clustering with a score of 0.308. Also, by getting a pretty good precision scores, the F1 score of 0.458 is also high for spectral clustering.

On the test data, the CR score is good on Fusion-Graph approach and the overall best F1 score is achieved on Fusion - Voting approach. This makes sense as these

Run	Dev Set			Test Set		
	P@10	CR@10	F1@10	P@10	CR@10	F1@10
1	0.681	0.217	0.329	0.696	0.221	0.335
2	0.783	0.231	0.357	0.779	0.219	0.342
3	0.780	0.278	0.410	0.777	0.268	0.399
4	**0.792**	0.311	0.447	0.788	0.301	0.436
5	0.784	0.282	0.415	**0.789**	0.273	0.406
6	0.779	**0.318**	**0.452**	0.774	0.298	0.430
7	0.781	0.279	0.411	0.785	0.273	0.405
8	0.787	0.314	0.449	0.782	0.309	0.443
9	0.759	0.281	0.410	0.764	0.274	0.403
10	0.773	0.317	0.450	0.777	**0.313**	**0.446**

Table 27. Precision, Recall and F1 Score for all runs. Number of Images retrieved is 10

methods are kind of ensemble different clustering algorithms and might have solved the high variance problem of a single algorithm.

Figure 8 shows the results for precision vs. cluster recall averages at 10 images.

Table 28, show the Precision, Cluster Recall and F1 Score for all runs. The Number of Images retrieved is 20. On the development data set. the cluster recall rate is high on Fusion Voting Based Clustering with a score of 0.562. Also, by getting a pretty good precision scores, the F1 score of 0.673 is also high for Fusion Graph Based Clustering.

On the test data, the CR score is good on Fusion-Graph approach performed the best based on CR Score and the net F1 Score. This makes sense as these methods are kind of ensemble different clustering algorithms and might have solved the high variance problem of a single algorithm.

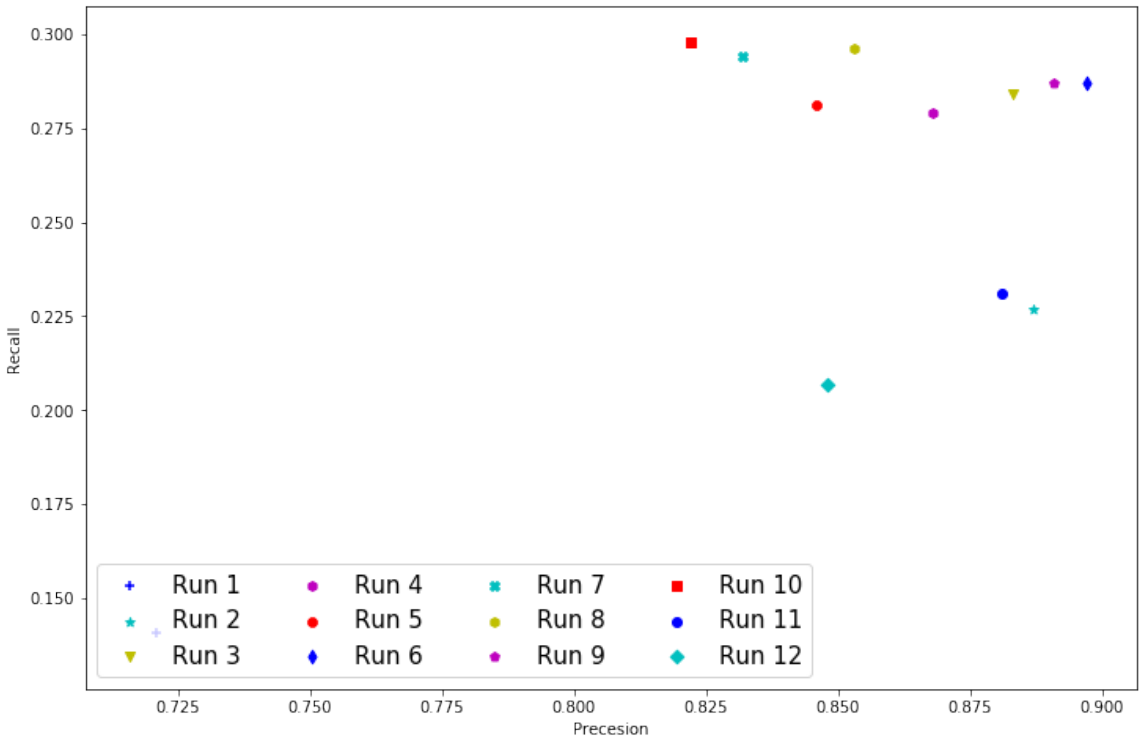

Figure 28. Precision and Recall values for 10 images retrieved on test set

Run	Dev Set			Test Set		
	P@20	CR@20	F1@20	P@20	CR@20	F1@20
1	0.653	0.245	0.356	0.641	0.257	0.366
2	**0.747**	0.273	0.400	0.739	0.271	0.396
3	0.731	0.451	0.558	**0.744**	0.443	0.553
4	0.722	0.488	0.582	0.711	0.478	0.571
5	0.728	0.465	0.568	0.733	0.456	0.574
6	0.729	0.511	0.601	0.735	0.499	0.54
7	0.721	0.475	0.573	0.741	0.470	0.575
8	0.727	0.513	0.602	0.729	0.512	0.602
9	0.741	0.485	0.586	0.732	0.482	0.581
10	0.723	**0.522**	**0.606**	0.728	**0.518**	**0.605**

Table 28. Precision, Recall and F1 Score for all runs. Number of Images retrieved is 20

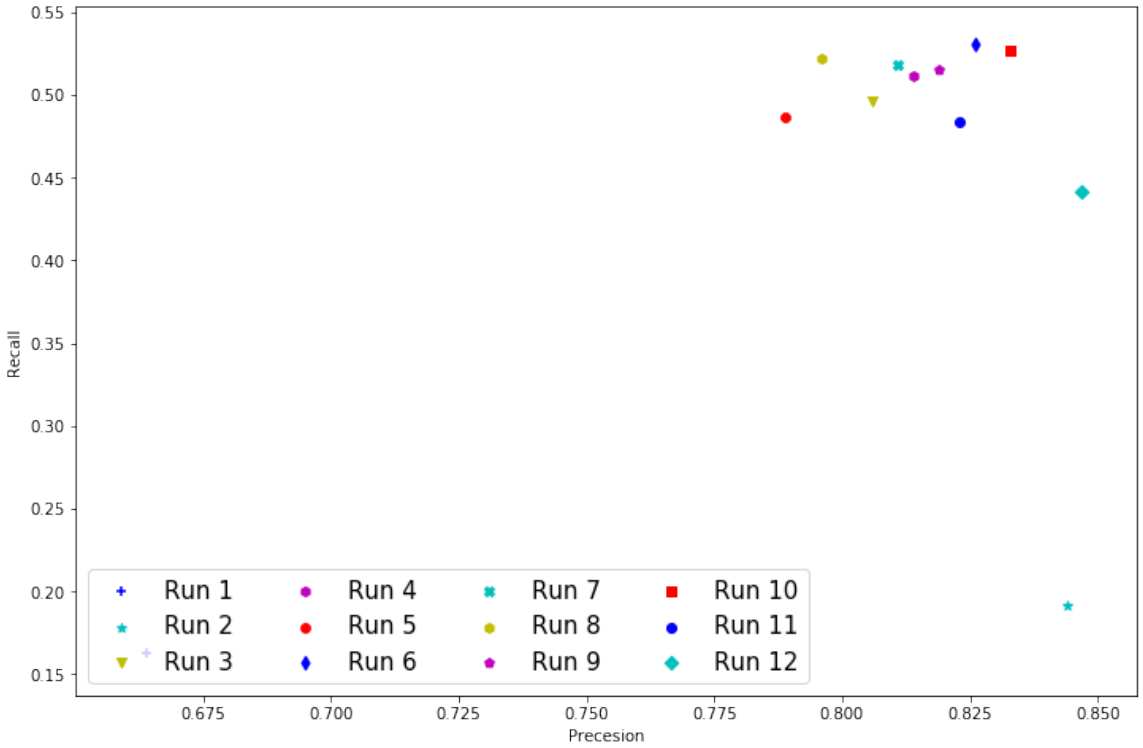

Figure 29. Precision and Recall values for 20 images retrieved on test set

6.4 Different Runs for Landmark Query Using Pre-filtering

In the previous section, we have seen different runs with out any prefilitering like removing blurred and facial images. Though the results are encouraging, we need to remove these kind of images taking quality of the images into consideration. The major problem we have in doing this is the number of training samples that are needed diminishes. This is also discussed in the data statistics section previously.

Below, is the results for 10 images retrieved. As you can see, there is not a big improvement by using the filters compared to initial results. There is a 0.004 F1 score improvement which is very minimal. The best F1 Score is obtained on Fusion Voting method with Facial Images removed.

Below image is the results for the top 20 images retrieved. The best result on the test set is obtained taking Fusion Graph based approach into consideration. We have obtained the score of 0.648 for the approach without any filter. Using filters, we have

Different Runs and their Settings				
Run	Pre-filtering	Image Diversification	Clusters	Diversification Features
1	None	Spectral	10	GLRLM
2	None	Fusion - Voting	10	NA
3	None	Fusion - Graph	10	NA
4	Face	Spectral	10	GLRLM
5	Face	Fusion - Voting	10	NA
6	Face	Fusion - Graph	10	NA
7	Face,Blur	Spectral	10	GLRLM
8	Face,Blur	Fusion - Voting	10	NA
9	Face,Blur	Fusion - Graph	10	NA
10	Face,Blur,Dist	Spectral	10	GLRLM
11	Face,Blur,Dist	Fusion - Voting	10	NA
12	Face,Blur,Dist	Fusion - Graph	10	NA

Table 29. Table showing settings of different runs

Run	Test Set		
	P@10	CR@10	F1@10
1	0.774	0.298	0.430
2	0.782	0.309	0.443
3	0.777	0.313	0.446
4	0.782	0.324	0.458
5	0.795	0.335	0.471
6	0.788	0.331	0.466
7	0.783	0.321	0.455
8	0.797	0.325	0.462
9	0.776	0.333	0.466
10	0.818	0.349	0.489
11	**0.822**	0.363	0.504
12	0.813	**0.366**	**0.505**

Table 30. Precision, Recall and F1 Score for all runs. Number of Images retrieved is 10

obtained the best score of 0.648 using Facial image filter. This is the same as the using without filters.

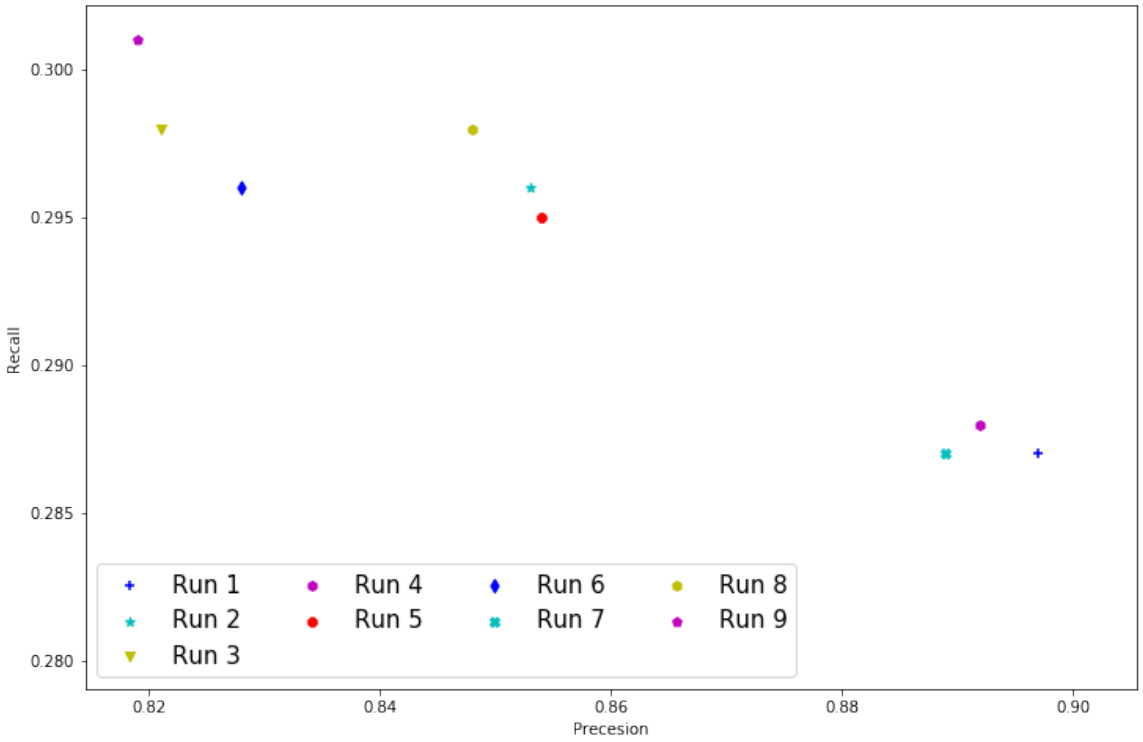

Figure 30. Precision and Recall values for 10 images retrieved on test set using filters

Run	Test Set		
	P@20	CR@20	F1@20
1	0.735	0.499	0.594
2	0.729	0.512	0.601
3	0.728	0.518	0.605
4	0.743	0.506	0.602
5	0.737	0.523	0.611
6	0.744	0.518	0.610
7	0.731	0.504	0.596
8	0.742	0.526	0.615
9	0.747	0.525	0.616
10	0.759	0.551	0.638
11	**0.761**	0.558	0.644
12	0.758	**0.561**	**0.645**

Table 31. Precision, Recall and F1 Score for all runs. Number of images retrieved is 20

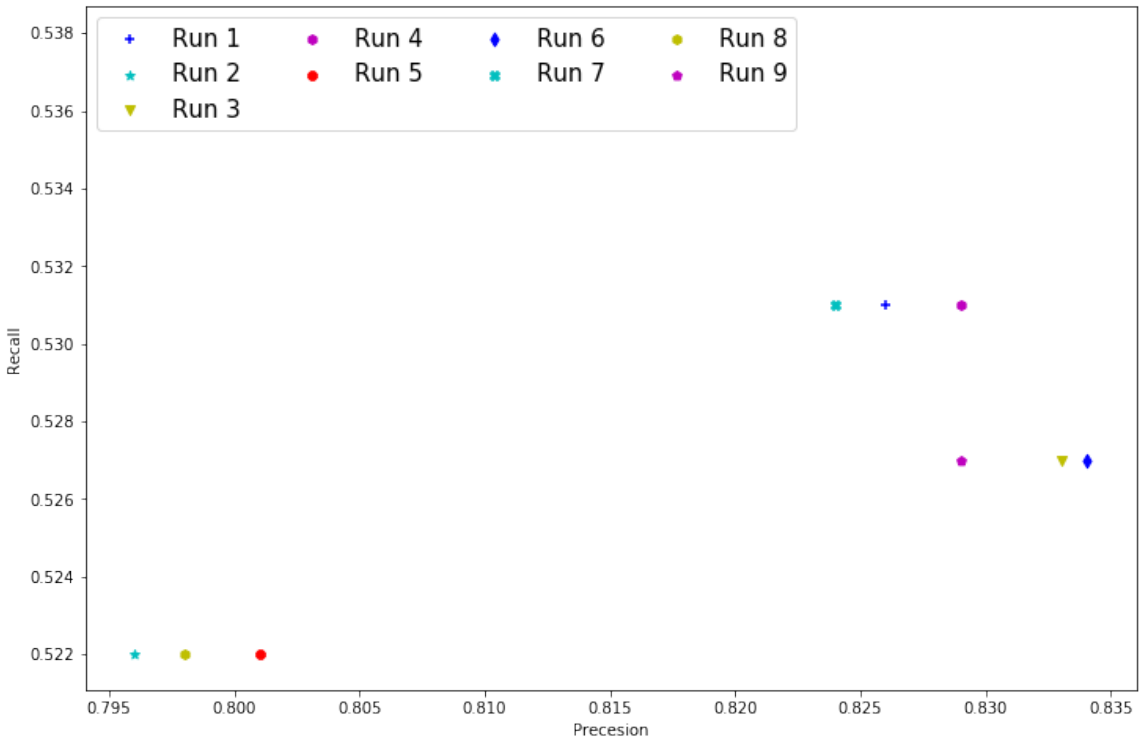

Figure 31. Precision and Recall values for 20 images retrieved on test set using filters

6.5 Comparison with Other Works

In this section, we will be comparing our best runs with the base line model and the other best results obtained in the competition. We will be using the our Spectral Clustering, Fusion Clustering approaches on 10 clusters in order to make a comparison. In this section we are comparing our results with the other works on P@10, CR@10, F1@10, P@20, CR@20, F1@20, P@30, CR@30, F1@30.

In the below table we are have taken the top runs and are comparing with our own methods.

The 1st row shows the results as obtained from Flickr Base Line Run. The next three rows show the top 3 runs obtained from the MediaEval Challenge - 2014. As

Different Runs and their Settings	
Run	Algorithm
1	Flickr Baseline
2	SOTON-WAIS run3
3	SocSens run1
4	CEA run2
5	Run6 - Spectral 10 Clusters
6	Run 8 - Fusion Voting 10 Clusters
7	Run 10 - Fusion - Graph 10 Clusters

Table 32. Table showing settings of different comparative runs

	10 images			20 images			30 images		
	P@10	CR@10	F1@10	P@20	CR@20	F1@20	P@30	CR@30	F1@30
1	0.756	0.325	0.449	0.728	0.515	0.578	0.719	0.655	0.657
2	0.815	**0.439**	**0.545**	0.778	0.619	**0.660**	0.741	0.721	0.701
3	0.733	0.429	0.520	0.748	**0.631**	0.659	**0.760**	0.722	0.708
4	0.782	0.422	0.531	0.730	0.626	0.649	0.725	**0.747**	0.707
5	0.818	0.349	0.489	0.759	0.551	0.638	0.748	0.692	0.705
6	**0.822**	0.363	0.504	**0.761**	0.558	0.644	0.755	0.705	**0.716**
7	0.813	0.366	0.505	0.758	0.561	0.645	0.752	0.699	0.710

Table 33. Precision, Recall and F1 Score for all runs - picked based on cluster size

you can see, the proposed models are under performing when F@10 and F@20 are taken into consideration but are outperforming when you consider F@30.

Table 34 shows the results when the images in each of the clusters are picked sequentially. Notice that the precision rates had dropped drastically when compared with the previous table. Also, the cluster recall rate improved (0.444 is the highest cluster recall score obtained when 10 images are retrieved for voting based fusion approach)

The figure 32 shows the precision results for all the algorithms compared. Our model outperforms most of the other models when Precision is taken into consideration. This can be attributed to the fact that we are preforming a fine tuning taking the top 100 relevant images in the initial step.

	10 images			20 images			30 images		
	P@10	CR@10	F1@10	P@20	CR@20	F1@20	P@30	CR@30	F1@30
1	0.756	0.325	0.449	0.728	0.515	0.578	0.719	0.655	0.657
2	**0.815**	0.439	**0.545**	**0.778**	0.619	**0.660**	0.741	0.721	0.701
3	0.733	0.429	0.520	0.748	**0.631**	0.659	**0.760**	0.722	**0.708**
4	0.782	0.422	0.531	0.730	0.626	0.649	0.725	**0.747**	0.707
5	0.702	0.397	0.487	0.717	0.589	0.627	0.723	0.688	0.685
6	0.689	**0.444**	0.520	0.699	0.604	0.628	0.734	0.703	0.650
7	0.711	0.431	0.516	0.734	0.616	0.650	0.727	0.719	0.702

Table 34. Precision, Recall and F1 Score for all runs - picked sequentially

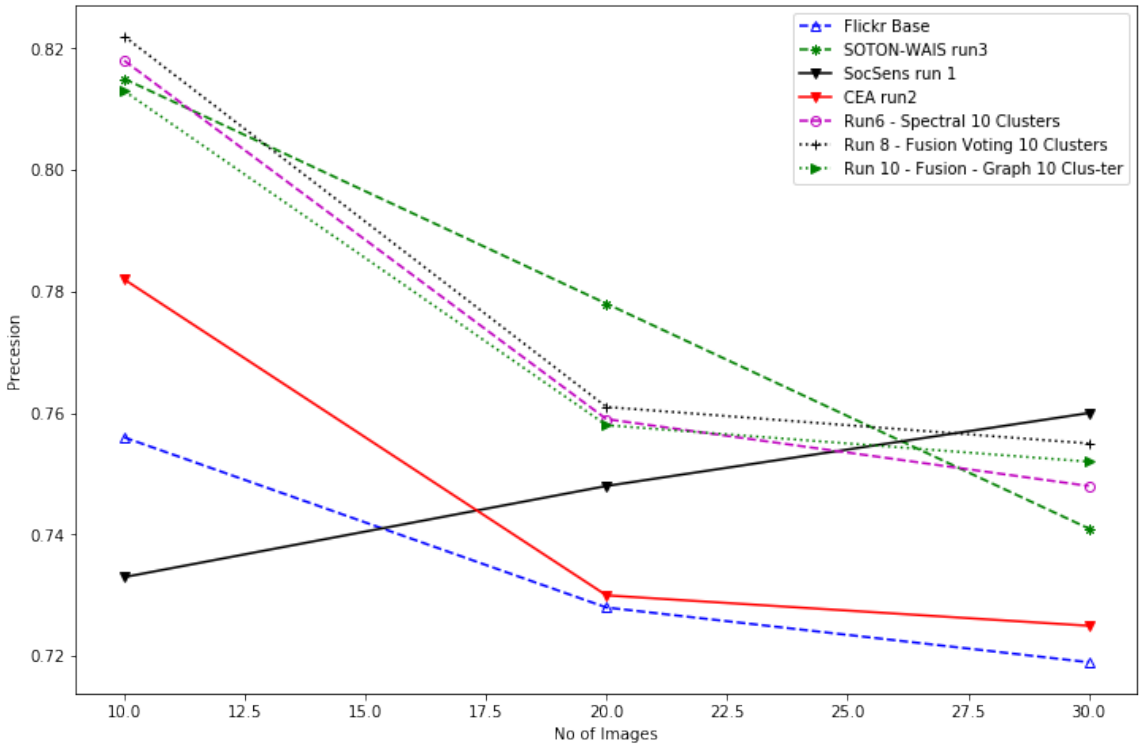

Figure 32. Precision plot for all 7 comparisons - Selection based on cluster size

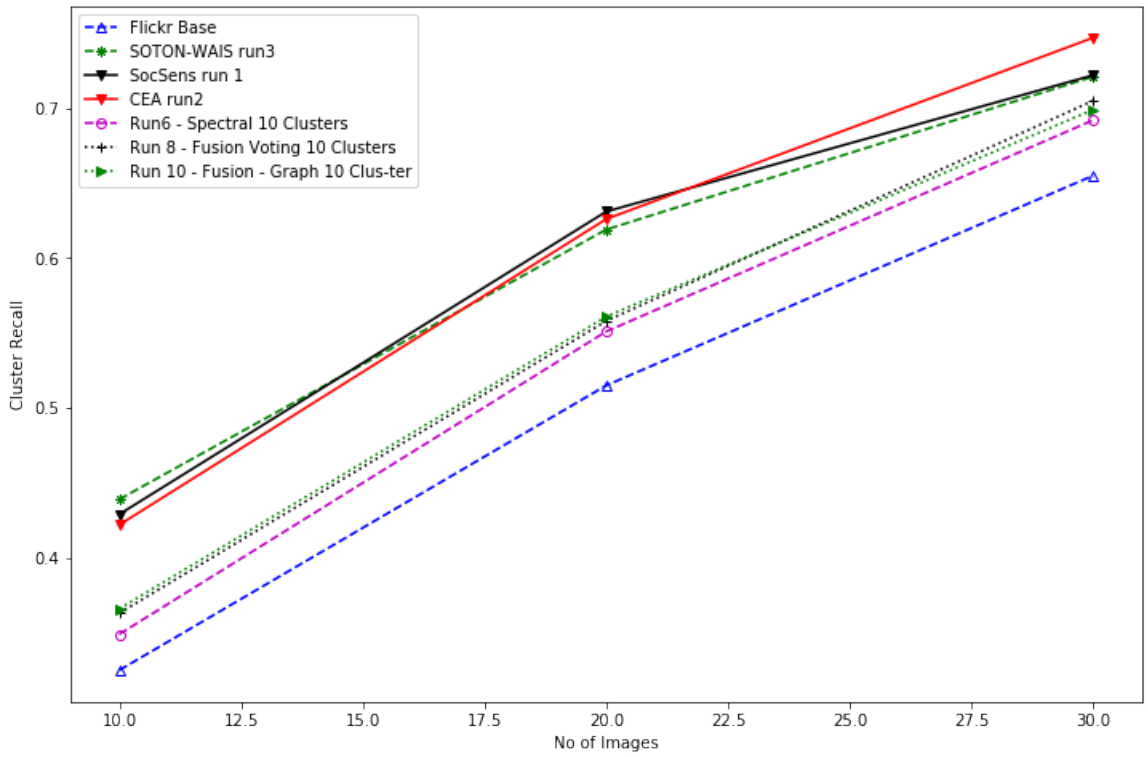

Figure 33. Cluster Recall plot for all 7 comparisons - Selection based on cluster size

The figure 33 shows the Cluster Recall scores for all of the algorithms. The Cluster Recall scores under performed in most of the runs. The reason can again be attributed to the fact that recall values in our case are inversely related the precision results. Secondly, we chose our results based on the cluster size. This is done in order to increase the precise results as taking images from smaller clusters might lead to outliers.

The figure 34 shows the F1 Score for all of our algorithms. The official F1 score in our challenge is F1@20. Our best F1 Score 0.644 in 7th amongst the top 30 results. Results showed that as the number of images retrieved increased, our algorithms worked better. As you can see in the figure, the F1 Score for 30 images outperformed the best algorithm.

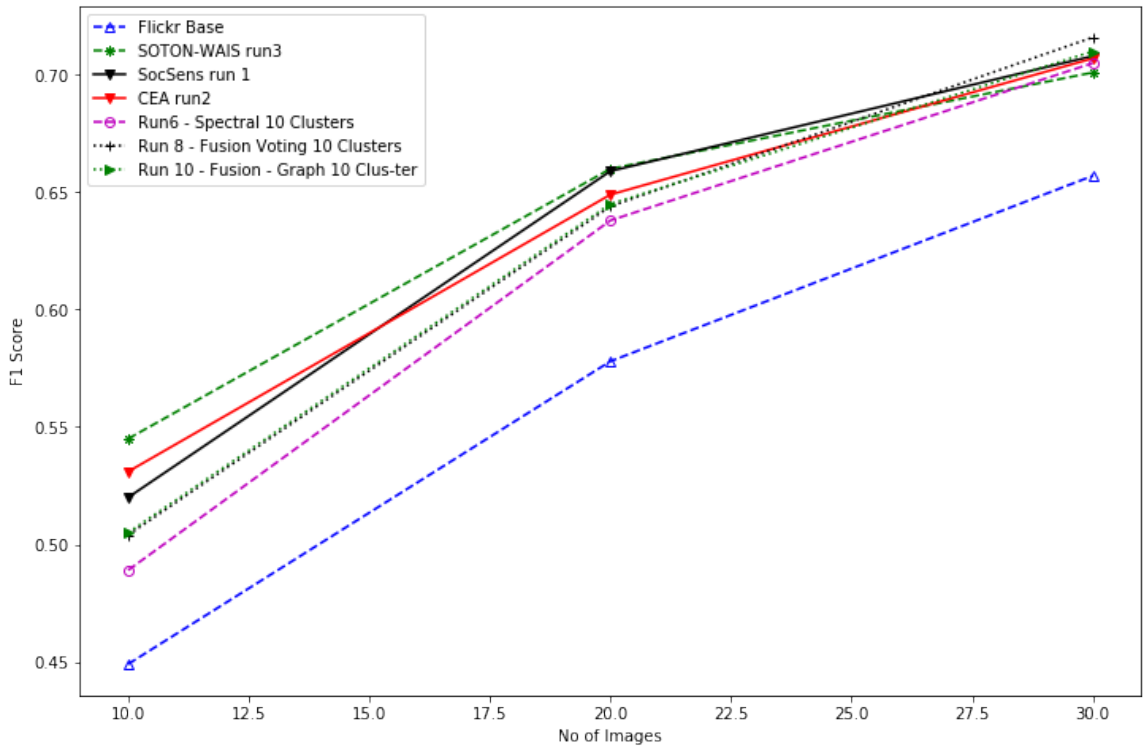

Figure 34. F1 Score plot for all 7 comparisons - Selection based on cluster size

Figures 35,36,37 shows the plots for the results obtained when we select nodes sequentially. Be can observe that the precision rate (Figure 35) has decreased drastically. This can be due to fact that there are cluster with 1 or 2 images (outliers) that might not have been filtered in the initial stages. These results get into our output results when we consider the images sequentially i.e one from each of the cluster until we get the required number of images needed.

Figure 36 shows the cluster recall rate plots. The recall rate has improved over the previous case (selecting images based on the size of the clusters). In the present scenario, we are selecting the images sequentially and so selecting a minimum of an

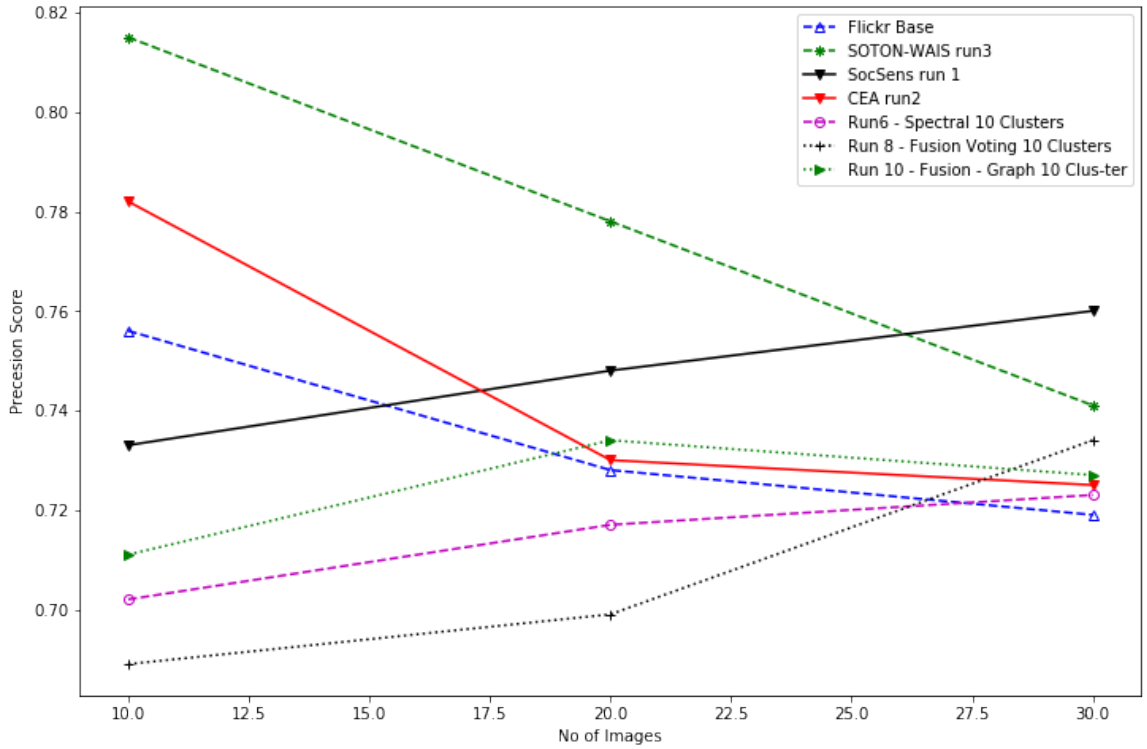

Figure 35. Precision plot for all 7 comparisons - Selection done sequentially

image from each of the clusters. This helps in diversifying our results better as that reflects in our cluster recall scores

Figure 37 shows the harmonic mean of precision and recall *i.e.*, F1 Score. It can be observed that the F1 Score obtained in this method are less than the previous (cluster size for picking the data item).

Figure 38, shows example 7 images obtained after the clustering step. From the results, we can see that different results are obtained with different lighting, angle (side view, top view) etc, some during the day time and some during the night, some in low lights etc.

Figure 39 shows the results for the query Agra Fort. There is still a bit of images

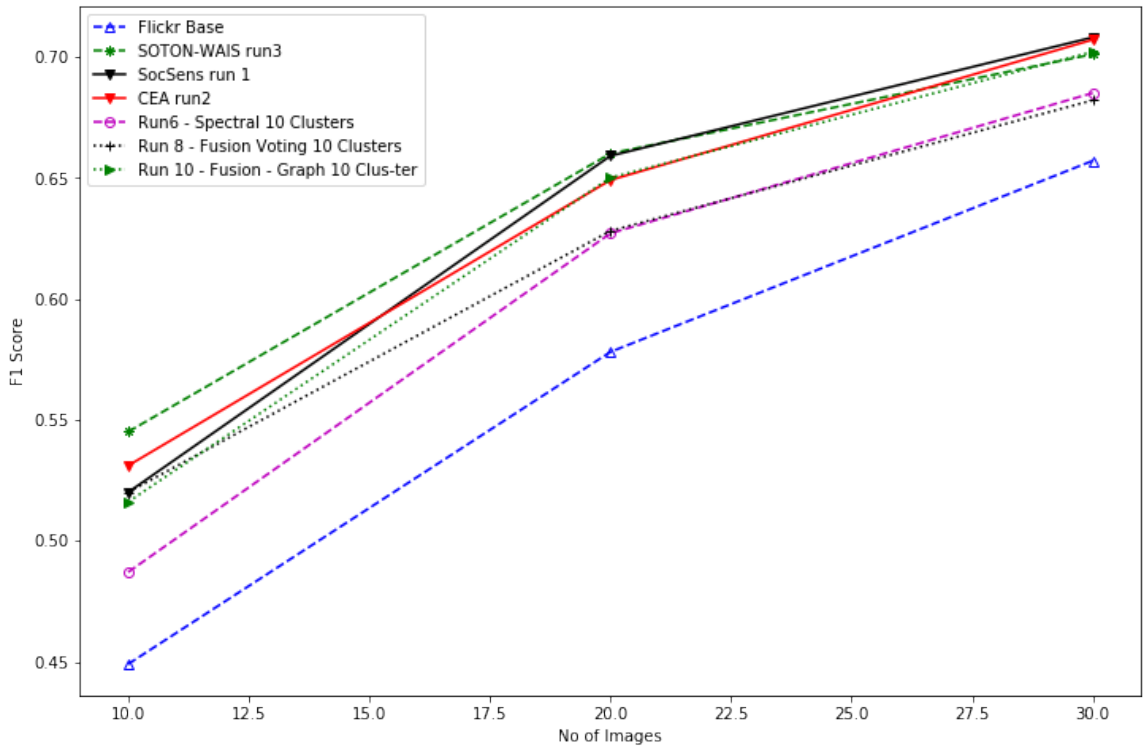

Figure 36. Cluster Recall plot for all 7 comparisons - Selection done sequentially

that are redundant. It can be observed that the first image in the second row and the last second in the same row are almost redundant. This is the biggest disadvantage of using this approach (ordering the clusters based on the size and picking the number of result based on the ratios)

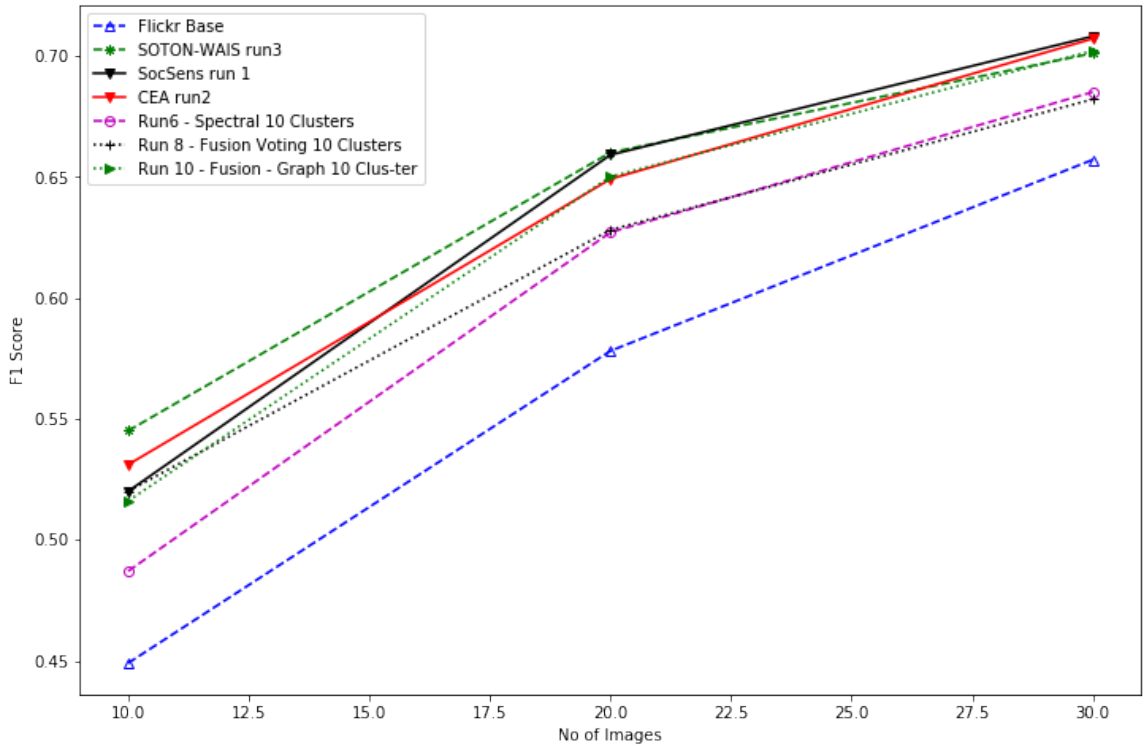

Figure 37. F1 Score plot for all 7 comparisons - Selection done sequentially

Figure 38. Example Results taken for Acropolis Athens, Neues Museum and Angkor wat

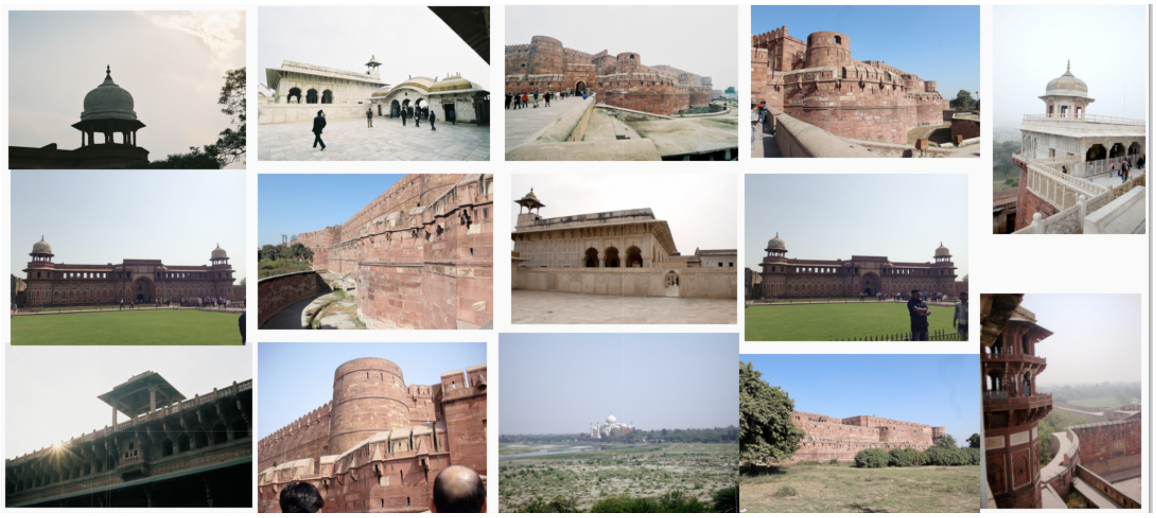

Figure 39. Results for the query - Agra Fort

Chapter 7

IMPROVEMENTS AND FUTURE WORK

The section discusses the possible improvements that can be done on the dataset and the model.

7.1 Improvements to Dataset

As discussed in the previous sections, one of the difficult faced in out thesis is the amount of images that are provided for each of the landmarks. One possible improvement could have been scraping more data. The problem with this is that there is no Image Hosting Service that is providing the data for free or providing relevant landmarks related data. A way to negate this is to scrape the data from different image hosting services. This would have helped us in handling the cases with no description, title and tags better which would have helped us in handing the relevant results better. This would have also helped us in using Deep Learning in order to solve our problem as deep learrning is known to improve as the amount of data increases.

Secondly, the relevancy of our results have drastically reduced for landmarks that are quite similar. Take the example of Altes Museum and Neues Museum as shown in Figure 23 and 24. Both these museums are located in Berin, Germany. The are situated beside each other and share a similar history - both of them have been bombed during the Word War 2 and were reconstructed in 2000's. It was very hard to find relevant results among them. Visually they are similar. Also, the results are

diverse as both of there contain antiquities collections that are quite similar. The only distinct textual metadata that can be used to differentiate them is to use their name. Most of the images that are provided didn't contain their name in either their title, tags or the description. The accuracy's that are achieved for both of these landmarks are 62 and 78 percent respectively. 62 percent is the least any landmark performed in our dataset. Looking further into the data, it is found that 26 percent of the results for Altes Museum belong to that of Neues Museum. Neues Museum performed slightly better competitively. Of the top 100 images only 11 images belonged to Altes Museum. Even then, 11 out of 22 wrongly classified images is equivalent to 50 percent of the results that are inaccurate. Further research needs to be done inorder to solve this problem. If location and similar kind of landmark is a reason, the sculptures present in the results also look similar. If you take a look into the Figure 23 and 24, we can observe that some images are similar (some are exactly the same). The reason can be due to the way people tagged the photos. Many of the images are wrongly tagged or confused. Overall, there are 4 images which are tagged in both Altes Museum and Neues Museum, 2 images from Altes Museum that actually belong to Neues Museum and a image vice versa. All these have contributed to a decrease in accuracy. Our future work needs to focus on improving it.

Thirdly, the images we have are got from Social Media. It has been uploaded my people who range from scholars to ones who aren't good with spellings and grammar. There are many images where the description has been wrongly spelled. Most of these words that are wrongly spelled are nouns or adjectives which have a high tf-idf score as they are unique to a particular image or location. Since there words are misspelled, they never contribute to the similarity measure that we compute to get the relevant

Figure 40. Sculptures related to landmark - Altes Museum

Figure 41. Sculptures related to landmark - Neues Museum

ranking. In future, work would be done in order to find words that are misspelled that could possibly improve our accuracy measures

7.2 Improvements on Query

In order to add the diversity to this query, we can add a multi-concept queries related to events and states associated with locations, e.g., "Altes Museum in winter", "Agra Fort in the night", etc. These queries are more complicated to solve and haven't been explored in our thesis. Since the relevant result accuracies are good for most of the single-concept queries, we can further improve on our search results and personalize them based on the user taste. For example, if a user wants to visit Agra Fort during the night, he would be more interested in the images that are taken in the than that are taken at any varied times.

7.3 Improvements in Methodology

One way to improve the relevant results is to combine both textual metadata and visual data in order to process the similarity measure. Even though we have tried exploring this in our thesis, it didn't give us better results that using only textual metadata. The problem with this approach is that we have a varied and diverse images for each of the landmarks that cannot be rightly classified. One way to solve this problem is to use the images from Wikipedia. Using the image from Wikipedia, we can access the images that are closest images from the dataset. Manually looking into the dataset, we can observe that the textual data in these images are more discriminating

between different landmarks. We believe that using this methodology, we can improve the final results.

7.3.1 Location Based Similarity to Re-rank Results

From the set of images, we can compute a similarity matrix taking all the images in a location into consideration. Using this matrix, we can find the most similar landmarks a given landmark belongs to. Using this, we can remove some of the unwanted results that come up in our relevant ranking of the results. Example can be between - Altes Museum and Neues Museum as mentioned previously.

7.3.2 Recompute Tf-idf with a Weightage to Title and Tags

Instead of computing the tf-idf on all the textual metadata evenly, weightage can be given to the title and title for each of the images. This will help in more efficient ranking of the results than giving equal weightage to all the 3 (title, tags and description).

Chapter 8

CONCLUSION

The idea of building this framework is to enable it to perform better on multiple datasets than on a single dataset. Solving a unsupervised problem is challenging as we are not sure of the parameters that needs to be considered and tuned inorder to attain a better performance. To get to the a right model, various experimentation and optimizations have be done. Various comparisons have been made in both finding the relevant and divergent results.

The main contributions of this work are :

- A novel fusion based clustering solution that can be used on different diverse image datasets without the problem of overfitting
- A novel approach "Max a Min" K Means clustering algorithm that solves the problem of initialization for K Means on a highly relevant data
- A model that can be extended to other divergent tasks of Aspectual Retrieval. Example: "extinct species" where the users might be more interested in diversity of relevant results

Other key contributions are as follows:

- Model obtained the best results for Precision Metric on 10,20 images retrieved
- Model that is also able to achieve the best F1 Score for 30 images retrieved
- Model that retrieves results on landmark description based queries along with the straightforward landmark name based queries

REFERENCES

Antonie, Maria-Luiza, Osmar R Zaiane, and Alexandru Coman. 2001. "Application of data mining techniques for medical image classification." In *Proceedings of the Second International Conference on Multimedia Data Mining,* 94–101.

Ayad, Hanan G, and Mohamed S Kamel. 2007. "Cumulative voting consensus method for partitions with variable number of clusters." *IEEE transactions on pattern analysis and machine intelligence* 30 (1): 160–173.

Boongoen, Tossapon, and Natthakan Iam-On. 2018. "Cluster ensembles: A survey of approaches with recent extensions and applications." *Computer Science Review* 28:1–25.

Bosch, Anna, Xavier Muñoz, and Robert Martı. 2007. "Which is the best way to organize/classify images by content?" *Image and vision computing* 25 (6): 778–791.

Cai, Deng, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. 2004. "Hierarchical clustering of WWW image search results using visual, textual and link information." In *Proceedings of the 12th annual ACM international conference on Multimedia,* 952–959.

Chopde, Nitin R, and Mangesh K Nichat. 2013. "Landmark based shortest path detection by using A* and Haversine formula." *International Journal of Innovative Research in Computer and Communication Engineering* 1 (2): 298–302.

Colas, Fabrice, and Pavel Brazdil. 2006. "Comparison of SVM and some older classification algorithms in text classification tasks." In *IFIP International Conference on Artificial Intelligence in Theory and Practice,* 169–178. Springer.

Dagli, Charlie K, Shyamsundar Rajaram, and Thomas S Huang. 2006. "Utilizing information theoretic diversity for SVM active learn." In *18th International Conference on Pattern Recognition (ICPR'06),* 2:506–511. IEEE.

Dang, Van, and W. Bruce Croft. 2012. "Diversity by Proportionality: An Election-Based Approach to Search Result Diversification." In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval,* 65–74. SIGIR '12. Portland, Oregon, USA: Association for Computing Machinery. doi:10.1145/2348283.2348296.

Datar, Mayur, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. 2004. "Locality-sensitive hashing scheme based on p-stable distributions." In *Proceedings of the twentieth annual symposium on Computational geometry,* 253–262.

Datta, Ritendra, Dhiraj Joshi, Jia Li, and James Z Wang. 2008. "Image retrieval: Ideas, influences, and trends of the new age." *ACM Computing Surveys (Csur)* 40 (2): 1–60.

Deng, Ting, and Wenfei Fan. 2014. "On the Complexity of Query Result Diversification." *ACM Trans. Database Syst.* (New York, NY, USA) 39, no. 2 (May). doi:10.1145/2602136.

Deselaers, Thomas, Tobias Gass, Philippe Dreuw, and Hermann Ney. 2009. "Jointly optimising relevance and diversity in image retrieval." In *Proceedings of the ACM international conference on image and video retrieval,* 1–8.

Dimitriadou, Evgenia, Andreas Weingessel, and Kurt Hornik. 2001. "Voting-merging: An ensemble method for clustering." In *International Conference on Artificial Neural Networks,* 217–224. Springer.

Domeniconi, Carlotta, and Muna Al-Razgan. 2009. "Weighted cluster ensembles: Methods and analysis." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2 (4): 1–40.

Fern, Xiaoli Zhang, and Carla E Brodley. 2004. "Solving cluster ensemble problems by bipartite graph partitioning." In *Proceedings of the twenty-first international conference on Machine learning,* 36.

Fred, Ana LN, and Anil K Jain. 2005. "Combining multiple clusterings using evidence accumulation." *IEEE transactions on pattern analysis and machine intelligence* 27 (6): 835–850.

Frossyniotis, Dimitrios, Minas Pertselakis, and Andreas Stafylopatis. 2002. "A multi-clustering fusion algorithm." In *Hellenic Conference on Artificial Intelligence,* 225–236. Springer.

Gao, Bin, Tie-Yan Liu, Tao Qin, Xin Zheng, Qian-Sheng Cheng, and Wei-Ying Ma. 2005. "Web image clustering by consistent utilization of visual features and surrounding texts." In *Proceedings of the 13th annual ACM international conference on Multimedia,* 112–121.

Genkin, Alexander, David D Lewis, and David Madigan. 2007. "Large-scale Bayesian logistic regression for text categorization." *technometrics* 49 (3): 291–304.

Haralick, Robert M, Karthikeyan Shanmugam, and Its' Hak Dinstein. 1973. "Textural features for image classification." *IEEE Transactions on systems, man, and cybernetics,* no. 6: 610–621.

Khan, Hina A, Marina Drosou, and Mohamed A Sharaf. 2013. "Dos: an efficient scheme for the diversification of multiple search results." In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management,* 1–4.

Marques, Oge, and Borko Furht. 2002. *Content-based image and video retrieval.* Vol. 21. Springer Science & Business Media.

McCallum, Andrew, Kamal Nigam, et al. 1998. "A comparison of event models for naive bayes text classification." In *AAAI-98 workshop on learning for text categorization,* 752:41–48. 1. Citeseer.

McGinty, Lorraine, and Barry Smyth. 2003. "On the Role of Diversity in Conversational Recommender Systems." In *Case-Based Reasoning Research and Development,* edited by Kevin D. Ashley and Derek G. Bridge, 276–290. Berlin, Heidelberg: Springer Berlin Heidelberg.

Ntoutsi, Eirini, Kostas Stefanidis, Katharina Rausch, and Hans-Peter Kriegel. 2014. "" Strength Lies in Differences" Diversifying Friends for Recommendations through Subspace Clustering." In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management,* 729–738.

Priyatharshini, R, and S Chitrakala. 2012. "Association based image retrieval: a survey." In *International Conference on Advances in Information Technology and Mobile Communication,* 17–26. Springer.

Raina, Rajat, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. 2007. "Self-taught learning: transfer learning from unlabeled data." In *Proceedings of the 24th international conference on Machine learning,* 759–766.

Rudinac, Stevan, Alan Hanjalic, and Martha Larson. 2013. "Generating visual summaries of geographic areas using community-contributed images." *IEEE Transactions on Multimedia* 15 (4): 921–932.

Santos, Rodrygo L. T., Craig Macdonald, and Iadh Ounis. 2015. "Search Result Diversification." *Found. Trends Inf. Retr.* (Hanover, MA, USA) 9, no. 1 (March): 1–90. doi:10.1561/1500000040.

Smeulders, Arnold WM, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. 2000. "Content-based image retrieval at the end of the early years." *IEEE Transactions on pattern analysis and machine intelligence* 22 (12): 1349–1380.

Taneva, Bilyana, Mouna Kacimi, and Gerhard Weikum. 2010. "Gathering and ranking photos of named entities with high precision, high recall, and diversity." In *Proceedings of the third ACM international conference on Web search and data mining,* 431–440.

Van Leuken, Reinier H, Lluis Garcia, Ximena Olivares, and Roelof van Zwol. 2009. "Visual diversification of image search results." In *Proceedings of the 18th international conference on World wide web,* 341–350.

Vee, E., U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. 2008. "Efficient Computation of Diverse Query Results." In *2008 IEEE 24th International Conference on Data Engineering,* 228–236. April. doi:10.1109/ICDE.2008.4497431.

Verma, Vikas. 2014. "Image Retrieval And Classification Using Local Feature Vectors." *arXiv preprint arXiv:1409.0749.*

Wang, Xin-Jing, Wei-Ying Ma, Qi-Cai He, and Xing Li. 2004. "Grouping web image search result." In *Proceedings of the 12th annual ACM international conference on Multimedia,* 436–439.

Wu, Zifeng, Chunhua Shen, and Anton Van Den Hengel. 2019. "Wider or deeper: Revisiting the resnet model for visual recognition." *Pattern Recognition* 90:119–133.

Xue, Hui, Songcan Chen, and Qiang Yang. 2009. "Discriminatively regularized least-squares classification." *Pattern Recognition* 42 (1): 93–104.

Zhu, Xiaojin, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. "Improving diversity in ranking using absorbing random walks." In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference,* 97–104.