



TAMPERE UNIVERSITY OF TECHNOLOGY

SANJEEV POUDEL CHHETRI

**WEB SCALE IMAGE RETRIEVAL BASED ON IMAGE TEXT
QUERY PAIR AND CLICK DATA**

Master's thesis

Examiner(s):

Professor Moncef Gabbouj

Professor Tommi Mikkonen

Dr. Iftikhar Ahmad

Examiner and topic approved by the
Faculty Council of the Faculty of Com-
puting and Electrical Engineering on 4th
June 2014.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree in Information Technology

SANJEEV POUDEL CHHETRI: Web Scale Image Retrieval Based on Image
Text Query Pair and Click Data

Master of Science Thesis, pages 57

Month and year of completion: February 2015 (Examiner and topic were approved in
the faculty council meeting on 4th June 2014)

Major: Software System

Supervisor: Dr. Ifikhar Ahmad

Examiner(s): Prof. Moncef Gabbouj, Prof. Tommi Mikkonen,

**Keywords: Image Retrieval, Relevance Evaluation, Data Partitioning, Face
Bank, Feature Extraction**

The growing importance of traditional text-based image retrieval is due to its popularity through web image search engines. Google, Yahoo, Bing etc. are some of search engines that use this technique. Text-based image retrieval is based on the assumption that surrounding text describes the image. For text-based image retrieval systems, input is a text query and output is a ranking set of images in which most relevant results appear first. The limitation of text-based image retrieval is that most of the times query text is not able to describe the content of the image perfectly since visual information is full of variety. Microsoft Research Bing Image retrieval Challenge aims to achieve cross-modal retrieval by ranking the relevance of the query text terms and the images. This thesis addresses the approaches of our team MUVIS for Microsoft research Bing image retrieval challenge to measure the relevance of web images and the query given in text form.

This challenge is to develop an image-query pair scoring system to assess the effectiveness of query terms in describing the images. The provided dataset included a training set containing more than 23 million clicked image-query pairs collected from the web (One year). Also, a development set was collected which had been manually labelled. On each image-query pair, a floating-point score was produced. The floating-point score reflected the relevancy of the query to describe the given image, with higher number including higher relevance and vice versa. Sorting its corresponding score for all its associated images produced the retrieval ranking for the images of any query.

The system developed by MUVIS team consisted of five modules. Two main modules were text processing module and principal component analysis assisted perceptron regression with random sub-space selection. To enhance evaluation accuracy, three complementary modules i.e. face bank, duplicate image detector and optical character recognition were also developed. Both main module and complementary modules relied on results returned by text processing module. OverFeat features ex-

tracted over text processing module results acted as input for principal component analysis assisted perceptron regression with random sub-space selection module which further transformed the features vector. The relevance score for each query-image pair was achieved by comparing the feature of the query image and the relevant training images. For features extraction, used in the face bank and duplicate image detector modules, we used CMUVIS framework. CMUVIS framework is a distributed computing framework for big data developed by the MUVIS group.

Three runs were submitted for evaluation: “Master”, “Sub2”, and “Sub3”. The cumulative similarity was returned as the requested images relevance. Using the proposed approach we reached the value of 0.5099 in terms of discounted cumulative gain on the development set. On the test set we gained 0.5116. Our solution achieved fourth place in Microsoft Research Bing grand challenge 2014 for master submission and second place for overall submission.

PREFACE

This work has been conducted at the Department of Signal Processing of Tampere University of Technology. This work has been funded as an on-going research project work at the department of signal processing.

I thank my colleagues at the MUVIS research group and the personnel of the Department of Signal Processing for providing such a pleasant and inspiring working atmosphere. In particular, I would like to thank Prof. Moncef Gabbouj for his continuous support and guidance. I would like to extend my gratitude to Dr. Iftikhar Ahmad and Prof. Tommi Mikkonen, for supervising this thesis to its completion.

I would like to thank all people who had helped me to complete my thesis. I would like to thank all the member of MUVIS research group for their support. I would also like to thank my friends Bishwa Prasad Subedi and Prakash KC for their support and encouragement during my thesis work.

Last but not least, I would like to thank my parents, grandparents and siblings for their Moral support throughout my studies. Finally, I would like to dedicate this thesis to my grandfather Mr. Pit Bahadur Budahthoki.

Tampere, 31st December 2014

Sanjeev Poudel Chhetri
Orivedenkatu 8, F 125
33720 Tampere
FINLAND
Tel. +358 41 706 9243

Table of Contents

1.	Introduction	1
1.1.	Information Retrieval	1
1.2.	Image Retrieval Problem.....	1
1.3.	Text-Based and Content-Based Image Retrieval	2
1.4.	Web Image Search	3
1.5.	MSR-BING Image Retrieval Challenge	4
1.6.	Objective and Scope of Thesis	4
1.7.	Thesis Organization.....	5
2.	Background	6
2.1.	Image Retrieval Approaches	6
2.1.1.	Text Based Image Retrieval.....	6
2.1.2.	Content Based Image Retrieval.....	7
2.1.3.	Hybrid-Based Image Retrieval.....	11
2.2.	Image Retrieval Systems	12
2.2.1.	Google Image Search	12
2.2.2.	PicSearch.....	13
2.3.	Feature Extraction	14
2.3.1.	Local Binary Patterns.....	14
2.3.2.	Colour Structure Descriptor	17
3.	Implementation	20
3.1.	System Overview	20
3.1.1.	Text Processing.....	21
3.1.2.	Features	22
3.1.3.	PCA-assisted Perceptron Regression with Random Subspace Selection ($P^2R^2S^2$)	23
3.1.4.	Face Bank.....	26
3.1.5.	Duplicate Image Detection.....	27
3.1.6.	Optical Character Recognition.....	29
3.1.7.	Merging	29
3.1.8.	Secondary Submissions and Other Considered Methods	30
3.2.	CMuvis Framework.....	31
3.2.1.	Master.....	32
3.2.2.	Worker	33
3.2.3.	Worker Library Dependencies	36
3.2.4.	Worker Class Diagram.....	38
3.2.5.	Class Responsibility.....	38
3.2.6.	Sequence Diagram	39
3.3.	Feature Extraction Framework.....	40
3.3.1.	Visual Feature Extraction Framework	40

4.	Experimental Results	42
4.1.	Dataset	42
4.2.	Dataset Properties.....	43
4.2.1.	Properties of Clicked Queries (Labels).....	43
4.2.2.	Properties of Clicked Images	44
4.3.	Construction of the Evaluation Datasets	45
4.4.	Construction of the Training Dataset	46
4.5.	Evaluation.....	46
4.6.	Partial Results on Individual Modules	47
4.7.	Final Parameter Setting	47
4.8.	Overall Results	48
5.	Conclusion and Future Works.....	49
5.1.	Conclusion.....	49
5.2.	Future Work	49
	REFERENCES.....	51

LIST OF FIGURES

Figure 1-1 Keyword-based Image Search.....	3
Figure 1-2 Query By Example-based Image Search.....	3
Figure 2-1 Text Based Image Retrieval	7
Figure 2-2 Content Based Image Retrieval	7
Figure 2-3 Examples of textures	10
Figure 2-4 Google image search interface	13
Figure 2-5 PicSearch advance image search.....	13
Figure 2-6 Calculating the original LBP code and a constant measure	15
Figure 2-7 Example of LBP histogram	16
Figure 2-8 The circular (8,1), (16,2) and (8,2) neighbourhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the centre of a pixel	16
Figure 2-9 LBPs in a circularly symmetric neighbouring set of rotation invariant uniform local binary patterns	16
Figure 2-10 HMMD Colour Space	18
Figure 3-1 System Overview.	20
Figure 3-2 P ² R ² S ² Flow Diagram.....	23
Figure 3-3 Examples of partitioning feature space and data samples.....	24
Figure 3-4 Examples of PCA-assisted perceptron regression over sample partitioning.....	25
Figure 3-5 Synthesized face feature vector of Barack Obama.....	27
Figure 3-6 Overview of relevance evaluation using the face bank.....	27
Figure 3-7 duplicate image detector architecture.....	29
Figure 3-8 Relevance Scale Range	30
Figure 3-9 CMuvis System Overview.....	32
Figure 3-10 Worker architecture.....	34
Figure 3-11 Worker Sub-System	35
Figure 3-12 Sample Feature XML	36
Figure 3-13 Class Diagram, Worker	38
Figure 3-14 Sequence Diagram, Worker	40
Figure 3-15 Fex module interaction with CMUVIS application	41
Figure 4-1 Examples of clicked queries with click counts	44
Figure 4-2 Examples of Clicked Images.....	45

LIST OF TABLES

Table 2-1 Overview of commonly used feature in image retrieval	9
Table 2-2 HMMD Colour Space Quantization for CSD.....	18
Table 2-3 ANMRR Results for CSD Using the HMMD Colour Space	19
Table 3-1 class responsibility, CWorkerController	39
Table 3-2 class responsibility, CStateMachine	39
Table 3-3 class responsibility, CMuvisClient	39
Table 3-4 class responsibility, ControllerEventHandler	39
Table 3-5 class responsibility, CMuvisCommsDataHandler	39
Table 4-1 Dataset Statistics	46
Table 4-2 Parameters values used in our master submission.....	48
Table 4-3 DCG scores for different versions of our system over the development and test sets	48
Table 4-4 Number of query-image pairs where each module was used when setting the final score.....	48

LIST OF ABBREVIATIONS

ANMRR	Average Normalized Modified Retrieval Rank
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CBIR	Content Based Image Retrieval
CMUVIS	Cloud Muvis
CNN	Convolutional Neural Network
CSD	Colour Structure Descriptor
DCG	Discounted Cumulative Gain
DID	Duplicate Image Detector
FeX	Feature Extraction
HCT	Hierarchical Cellular Tree
HMMD	Hue-Max-Min-Diff
HTTP	Hypertext Transfer Protocol
ICME	International Conference on Multimedia and Expo
LBP	Local Binary Pattern
LTR	Learning To Rank
MB-LBP	Multi-Block Local Binary Patterns
MPEG	Motion Picture Coding Expert Group
MSR	Microsoft Research
NDCG	Normalized Discounted Cumulative Gain
OCR	Optical Character Recognition
$P^2R^2S^2$	Principal Component Analysis Assisted Perceptron Regression with Random Sub-Space Selection
PCA	Principal Component Analysis
PCs	Personal Computers
PQ	Progressive Query
TBIR	Text Based Image Retrieval
UI	User Interface
WWW	World Wide Web
XML	Extensible Markup Language

1. INTRODUCTION

The World Wide Web is increasing its presence and impact to our lives in various aspects. WWW is connecting different devices (mobile phone, tablets, PCs, laptops etc.) and making easy to share media items from different devices. With the prevalence of digital cameras and the Internet, there are more and more digital images on the Web. Huge amount of images are available online through various image/video sharing websites such as Flickr, YouTube, and Facebook. According to a global estimation the applications like Flickr has more than 6 billion photos, likewise YouTube has approx. 690 million videos and on the other hand Facebook contains more than 2 billion photographs [71]. The explosion of digital images necessitates effective and efficient image retrieval techniques. This chapter presents the information retrieval, image retrieval, web image search, MSR Bing challenge, scope, and outline of thesis.

1.1. Information Retrieval

In the past decade, with the expansion of multimedia technologies and the Internet, more and more information has been published in digital form. In the meanwhile, much of the information available in older books, journals and newspapers has been digitized. Music, images, satellite pictures, books, newspapers, and magazines have been made accessible for computer users. Internet enables the users to search and retrieve this vast information. The large information available about a given topics creates complex challenges for the user of World Wide Web in locating relevant information. Most users are aware of their needed information but the uncertainty of the location of information is a problem among them. In this case search engine has been their facilitator and assist them to find the location of the required and relevant information.

1.2. Image Retrieval Problem

In today's modern age, virtually all the spheres of human life including commerce, government, academics, hospitals, criminal offense prevention, surveillance, engineering, architecture, news media, fashion, graphic design, and historical research use images for efficient and informative services. An image database is a system comprise of huge collection of images where image data are integrated and stored [22]. Image data include information extracted from images by automated or computer assisted image analysis.

The police maintain image database for various categories such criminals, crime scenes, and stolen items. In the medical profession, image database exist for X-rays and scanned images, which are useful for diagnosis, monitoring, and research purposes. Im-

age database exists for design projects, finished projects, and machine parts which are essential for architectural and engineering design. The journalists create image databases for various events and activities such as sports, buildings, personalities, national and international events, and product advertisements, which are the essentials of publishing and advertising [45]. Images are more expressive than words. Browsing an image for identification is possible in a small collection of images. However, this is not the case for large and varied collection of images, where the user encounters the image retrieval problem. Due to explosive growth of digital images, effective and efficient retrieval techniques of images are needed. There is a problem when searching and retrieving for images that is relevant to a user requirement in a large image collection [15] [52]. In order to solve this problem the text-based and content-based image retrieval techniques are adopted for search and retrieval in an image database.

1.3. Text-Based and Content-Based Image Retrieval

The text-based image retrieval (TBIR) can be tracked back to 1970's [78]. In text-based retrieval, images are indexed using keywords, subject headings, or classification codes, which in turn are used as retrieval keys during search and retrieval [59]. Text-based retrieval techniques are simple and quick, but are non-standardized as different users employ different keywords for annotation, making them subjective [68]. Manual Annotation of images is a weighty and expensive task for large image databases and is often subjective, context-sensitive and incomplete [12]. Google [79], Yahoo [90], Bing [83] Image search engines are example of systems using text-based approach. Even though, these search engines are fast and robust but sometimes they fail to retrieve relevant images [44].

In the early 90's, manual annotation approach became more difficult because of emergence of large-scale image collections. To overcome the difficulties faced by text-based image retrieval, content-based image retrieval (CBIR) was proposed. CBIR systems utilize only visual features, such as colour, texture and shape to retrieve similar images. In general, the bottleneck to the efficiency of CBIR is the semantic gap between the high level image interpretation of the users and the visual feature stored in the database for indexing and querying. In other words, there is a difference between what image features can distinguish and what people perceives from the image [74]. When "search" become one of the most frequently used applications, "intent gap", the gap between query expression and user's search intents, emerged. MSR-Bing image retrieval challenge leverages the click data to bridge the semantic and intent gap using image dataset "Clickture" [72].

Clickture is a large-scale image dataset with real-world click-data generated from Microsoft Bing image search engine. The Clickture-lite version of the dataset is used in this challenge. This dataset was sampled from one-year click log of Bing image search engine. The data is organized by triads of queries, images and clicks as $Clickture = \{K, Q, C\}$, where a query (Q)-image (K) pair is coupled with the number

of clicks (C) from the search results. Click count can be seen as an indicator of the relevance of an image-query pair. In general, more clicks imply higher relevance between the query and image. The previous attempt to improve the ranking of image search based on click data includes top query modelling, image annotation by query modelling and rank learning [71]. Clickture is promising dataset to solve multiple problems in computer vision, including image retrieval, auto-annotation and image classification.

1.4. Web Image Search

Image search is becoming an increasingly important topic to facilitate access to the rapidly growing collections of images on the web. There are two main schemes for searching images on the Web i.e. keyword-based and query by example-based scheme. In the keyword-based scheme, images are searched for by a query in the form of a textual keyword. The query comprises of one or multiple keywords specified by user. Keyword-based scheme is illustrated in Figure 1-1.



Figure 1-1 Keyword-based Image Search

In the query by example-based scheme, images are searched for by a query, which is, an image, either specified by a URL or uploaded by users. Query By Example-based scheme is illustrated in Figure 1-2.



Figure 1-2 Query By Example-based Image Search

1.5. MSR-BING Image Retrieval Challenge

The challenge is about web scale image retrieval. The participants need to develop an image-query system through this challenge to assess the effectiveness of query terms (labels) in describing the images crawled from the web for image search purposes. On each image-query pair a floating-point score need to be produced. The floating-point score reflects the relevancy of use of query to describe the given image, with higher number indicating higher relevance and vice-versa. Sorting corresponding score for all its associated images produces the retrieval ranking of any query. [85]

The challenge is based on the Clickture dataset [72] which is by-product of Bing image search. Clickture reflects common user's searching and consuming interest as compared with other manually labelled dataset. It covers the semantics (textual queries) that people desire to search in daily life. The aim of the challenge is to leverage the massive amount of click data generated from image search engines to facilitate image recognition and search. The task is to find a relevance score of any given image-query pair based on the model, index, or any form of knowledge representation, which is built upon the Clickture dataset [72]. This challenge is much closer to real-world applications [73].

This is the second time that Microsoft Research in partnership with Bing organized MSR-Bing challenge on Image Retrieval. The results of the first challenge can be found in [82]. This year, six teams successfully submitted both prediction results and papers among the 11 registered teams. Our MUVIS team took part in both first and second challenge. This year, our solution achieved fourth place for master submission and second place for overall submission.

Our team combines a text-processing module with a module performing PCA-assisted perceptron regression with random sub-space selection ($P^2R^2S^2$). $P^2R^2S^2$ uses Overfeat (a CNN based representation) as a starting point and transforms them into more descriptive features via unsupervised training. The relevance score for each query-image pair is obtained by comparing the feature of the query image and the relevant training images. We also use Face bank, duplicate image detection, and optical character recognition to lift our evaluation accuracy.

1.6. Objective and Scope of Thesis

The objective of this thesis is to describe system designed and developed by MUVIS team in MSR-Bing Image retrieval Challenge at ICME 2014 based on Clickture dataset [72]. The task of MSR-Bing Retrieval Challenge is to find the relevance between text query and its returned image list, and is useful for result re-ranking in image search system. It helps to build up the link between short phrases/sentences and image. This thesis covers:

- a. Description of the System developed by MUVIS team for MSR-Bing Challenge.
- b. Calculation of relevance and confidence score for each image-query pair.

- c. Use of CMUVIS framework for feature extraction and finding exact match images.

1.7. Thesis Organization

This thesis is organized in five chapters. The first chapter gives brief introduction on web scale image retrieval, content-based image retrieval, web image search and MSR-BING image retrieval challenge. This chapter also provides insight about objective and scope of the thesis. Chapter two is focused on background knowledge required to understand the thesis. The descriptions of image retrieval, techniques of image retrieval, feature extraction used in this thesis are also depicted in this chapter. Chapter three gives detailed description of the proposed system. It describes different components and working principle of the system based on the theoretical foundations formed in Chapter two. This chapter also contains description of Cloud MUVIS (CMUVIS) framework, which was used for extracting features during the challenge. In chapter four, we show some experiment result based on provided dataset. And finally chapter 5 contains the summary of the contributions of this work, possible scope for the work in future and final conclusion.

2. BACKGROUND

This chapter develops the foundation for the key concepts used in this thesis. It discusses the theoretical details of these concepts. An image retrieval system is a computer system for searching, browsing, and retrieving images from a large image database of digital images [54]. It is growing research field in information technology with many real-world applications since 1970s [78]. Image retrieval systems are useful for many applications, ranging from art galleries and museum archives to picture collections, criminal investigation, medical and geographical databases. This chapter deals with different types of image retrieval systems, feature extraction, and also explanation of basic local features i.e. LBP [65], CSD [9] used in the challenge.

2.1. Image Retrieval Approaches

In general, an image is a representation of a real object or scene. The size of digital image collection is growing fast with the rapid development and improvement of the Internet, availability of image capturing device such as digital cameras, image scanners. Efficient and effective image searching, browsing and retrieval tools are required by users from various domains, including remote sensing, fashion, crime prevention, publishing, medicine, architecture, etc. For this purpose, many general purpose image retrieval systems have been developed. Some of them are: text-based, content-based and hybrid method [5].

2.1.1. Text Based Image Retrieval

Text based image retrieval system is used in almost all public web image retrieval systems today [7]. This approach uses the text associated with an image to determine what the image contains. This text can be text surrounding the image, the image's filename, a hyperlink leading to the image, an annotation to the image, or any other piece of text that can be associated with image [31]. Google [79], Yahoo [90], and Bing [83] Image search engines are example of systems using this approach. Although these search engines are fast and robust, they sometimes fail to retrieve relevant images. Figure 2-1 shows a text based image retrieval method. The pros and cons of text based image retrieval are given below [18]:

Pros:

- Easy to implement.
- Fast retrieval.

- Web image search (surrounding text).

Cons:

- Manual annotation is impossible for a large database.
- Manual annotation is not accurate.
- Polysemy problem (more than one object can refer by the same word).
- Surrounding text may not describe the image.
- A picture is worth a thousand words.

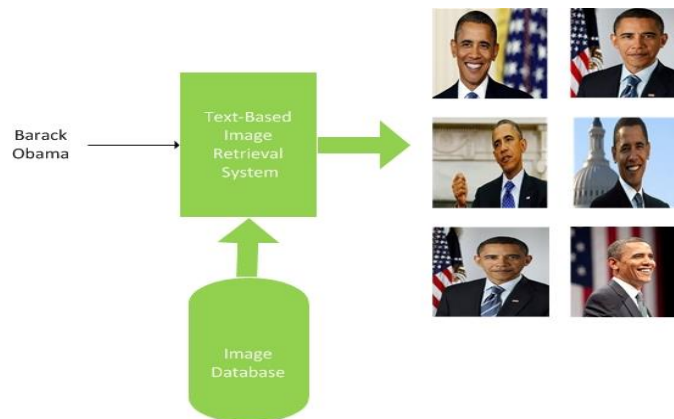


Figure 2-1 Text Based Image Retrieval

2.1.2. Content Based Image Retrieval

The term content-based image retrieval was originated in 1992 by T. Kato to describe experiments into automatic retrieval of images from a database, based on the colours and shape present [24].

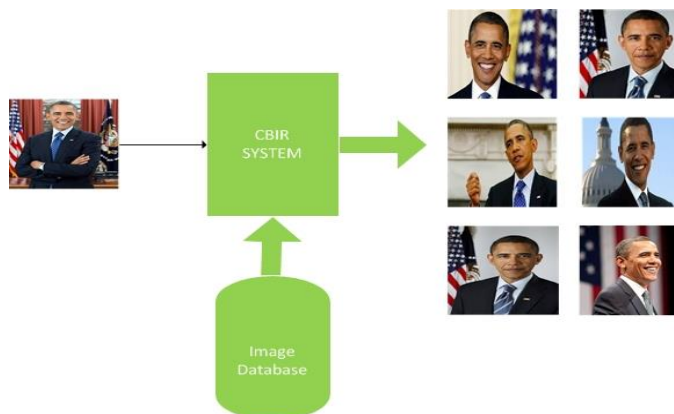


Figure 2-2 Content Based Image Retrieval

Since then, it has been used as alternative to text based image retrieval. IBM was the first, who take an initiative by proposing query-by image content (QBIC) [20]. CBIR is based on extraction of the low-level features of images, such as colour, texture and shape, to support visual queries in a spontaneous way with content descriptor [6]. Figure 2-2 shows the content-based image retrieval method. The pros and cons of content-based image retrieval are given below [18] [39]:

Pros:

- Visual features, such as colour, texture and shape information, of images are extracted automatically.
- Similarities of images are based on the distance between features [38].

Cons:

- Semantic gap.
- Querying by example is not convenient for a user [8].
- Too much of effort is needed to yield good results [39].

CBIR involves the following four parts in system realization: data collection, build up feature database, search in the database, process and sort the results [46].

1) Data collection

Data gathering is a systematic approach to gather information (images) from a variety of sources. For example, Web image crawler, which is used to collect a multitude of images from websites. In the case of Bing challenge, MSR and Bing provide development package for the participants. Development package contains training and development dataset intended for local debugging and evaluation.

2) Extract feature database

Features such as colour, texture, etc. are used to describe the content of the image. In this step, visual information is extracted from the images and saves them as feature vectors in a database.

3) Searching in the database

The CMUVIS framework extracts the feature of query image and compares the feature vector of query image against all feature vectors in the database to calculate the dissimilar distance. It returns the several related images with the minimum similar distance. The different distance methods available for measuring similarity are: Euclidean distance [3], Mahalanobis Distance [1], Canberra Distance [84]. Euclidean distance is used by CMUVIS framework for measuring similarity.

4) Process and sort the results

Sort the image obtained from searching due to the similarity of features, and then return the retrieval images to the users.

The different types of CBIR Systems are given below:

1) Region based image retrieval

The Netra developed in the UCSB Alexandria Digital Libraray (ADL) project [69] and Blobworld [11], developed at UC-Berkeley are two earlier region based image retrieval systems [48]. Region based image retrieval (RBIR) systems partitions an image into a number of homogenous regions and extract local features for each region. The features of the regions are used to represent and index images in RBIR. The user supplies a query object by selecting a region of a query

image. The similarity measure is computed between features of region in the query and a set of features of the segmented regions in feature database. Finally, the system returns a ranked list of images that contain the same object.

2) Object based image retrieval

Object based image retrieval systems retrieve the images from the database based on the appearance of physical objects in those image [44]. These objects can be apple, flowers, lion, dog, airplane or any other objects that the user wishes to find. One common way to search the objects in images is to first partition the images in database into multiple segment and then compare each segmented region against a region in some query image presented by user [13].

3) Example based image retrieval

In this type of image retrieval, the user gives the sample image and then system uses it as a base for the search. The system finds the images that are related to the base image based on visual features.

4) Recognition based image retrieval

The first Document Image Retrieval in digital libraries was based on the recognition-based paradigm, where document image analysis techniques (and mostly OCR packages) were used to recognize the informative content in the documents to be archived [62]. Citation analysis, Hand recognition, Layout recognition and Born-digital documents are some of the retrieval methods based on recognition based image retrieval [62].

5) Feedback based image retrieval

System display sample of pictures and asks rating from the user. Using these rating, system re-queries and repeats until the correct image is found.

Feature Extraction

Feature (content) extraction is the basis of content-based image retrieval. Typically the visual features can be classified as primitive features and domain specific features.

- The primitive features, which include colour, texture, and shape.
- Domains specific, which are application-specific and may include, for example human faces and fingerprints.

Commonly used feature extraction method in image retrieval is shown in Table 2-1.

Table 2-1 Overview of commonly used feature in image retrieval

Features	Feature extraction method to extract image
Color	Color Histogram, Color Moments, Color-Covariance matrix, Color Coherence Vector, Color Correlation, Color Structure Descriptor
Texture	Co-occurrence matrix, Tamura Feature, Gabor Transform, Wavelet Transform and Fourier Transform
Shape	Segmentation, Edge Detection, Fourier Descriptor and Moment Invariants

The remainder of the section will concentrate on those general features, which can be used in most applications.

Colour

The colour is one of the most widely used low-level visual features in CBIR systems. It is invariant to image size and orientation. Colour feature used in this thesis include YUV, HSV, and CSD. First a colour space is used to represent colour images. The gray level intensity is represented as the sum of red, green and blue gray level intensities for RGB space [2]. In image retrieval applications, colour intensities are usually encoded into a histogram that spans over the whole pixels of an image [30]. A colour histogram represents the frequencies of every intensity colour in the image. Computationally, the colour histogram is calculated by counting the number of pixels for each colour.

Similarity metrics are used to measure the distance between the query image and database image while retrieving images based on their colour histogram. Histograms Intersection Distance [41], Euclidean Distance [3], Mahalanobis Distance [1], Histogram Quadratic Distance [26] are some of the distance method to compute similarity in image retrieval. The image with smallest similarity value with respect to other is favoured over other images in the database.

Variety of colour spaces includes, RGB, HSI, LUV, YCrCb, HSV (HSL), and the hue-min-max-difference (HMMD) [43]. Common colour features or descriptor in CBIR systems include, colour-covariance matrix, colour histogram, colour moments and colour coherence vector. The Colour Structure Descriptor (CSD) [9] represents an image by both the colour distribution of the image or image region, and the local structure of the colour.

CSD are used for duplicate image detector (Section 3.1.5) to identify whether query image is a duplicate or near duplicate of an associated training images return by text processing module.

Texture

Texture is another important visual attribute useful for retrieving image. It refers to the visual patterns that have property of homogeneity that do not result from presence of only a single colour or intensity [29] [63]. Figure 2-3 shows a few types of textures. The textural feature opted in this thesis include Local Binary Patterns (LBP) [65]. Gabor wavelet feature are also used in texture feature extraction. There exist different approaches to represent and extract textures. Directional features are extracted to capture image texture information. Wavelets and Gabor filters are some of the most common measures for capturing the texture of images. The role of texture measures is to retrieve the image or image parts characteristics with reference to the changes in certain directions and the scale of the images. This is most useful for region or images with homogeneous texture [56].

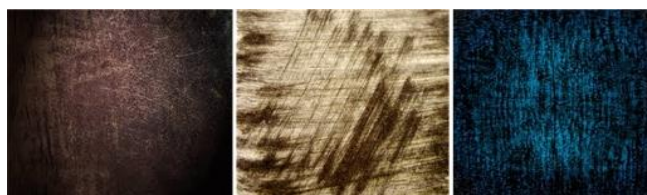


Figure 2-3 Examples of textures

The texture analysis methods can be divided into statistical, structural and spectral approaches.

- 1) “Statistical techniques which include the popular co-occurrence matrix, Fourier power spectra, shift invariant principal component analysis (SPCA), Tamura feature, Multi-resolution filtering technique such as Gabor and wavelet transform, characterize the texture by statistical distribution of the image intensity” [19]. This is the important techniques for texture classification.
- 2) Structural techniques characterize texture as being composed of texture elements. These texture elements are arranged regularly on a surface according to some specific arrangement rules.
- 3) Spectral approach includes Fourier transform of an image that gives textual description and then group the transformed data in a way that it gives some set of measurements.

Similar to CSD used in Bing challenge, LBP are also used for duplicate image detector (for more details check Section 3.1.5).

Shape

Shape is also one of the important attribute of an image. In general, the shape representation can be divided into two categories, region-based and boundary-based.

- 1) Region-based: Shape representation uses the entire shape region. This is done by describing the considered region using its internal characteristics i.e., the pixels contained in that region.
- 2) Boundary-based: Shape representation only uses the outer boundary of the shape. This is done by describing the considered region using its external characteristics i.e., the pixels along the object boundary.

In the late years, outer boundaries of the shape were only used while the current uses the entire shape region [77].

The most successful representatives for these two categories are Fourier descriptor and moment invariants.

- 1) Fourier descriptor: The main motive of Fourier descriptor is to use the Fourier transformed boundary as the shape feature.
- 2) Moment invariants: The main motive of moment invariant is to use region-based moments, which are invariant to transformations as the shape feature.

In our proposed system for Bing Challenge, we are not using any shape features.

2.1.3. Hybrid-Based Image Retrieval

A recent trend for image search is to fuse the two basic techniques of Web images, i.e., context (usually represented by the keywords) and visual features for retrieval [53]. The better result can be achieved only by making joint between existing textual context and visual features [27]. The simple approach for this method is based on counting the fre-

quency-of-occurrence of words for automatic indexing. The second approach takes a different stand and treats images and texts as equivalent data. It attempts to discover the correlation between visual features and textual words on an unsupervised basis, by estimating the joint distribution of features and words and posing annotation as statistical inference in a graphical model [5].

These approaches usually learn the keywords correlation according to the co-occurrence of keywords in the training set or the hierarchy of lexicon. However, for the scenario of annotating Web images or semantic meaning of keywords such as synonyms, the keywords correlation estimation is more subtle and complicated in some extent [23]. This is the reason behind not being able to combine traditional text-based and content-based image retrieval for dealing with the problem of image retrieval on the Web [5].

To improve image retrieval technique, different features of content-based image retrieval can be combined. Hybrid features can also be used to enhance the retrieval technique i.e. colour and texture, texture and shape based hybrid approach etc.

2.2. Image Retrieval Systems

Since the early 1990's, content-based image retrieval has become an active research area [78]. Many commercial and research oriented image retrieval systems have been built since then. Almost all of the image retrieval systems support one or more of the following options [58].

- Random browsing
- Search by example
- Search by sketch
- Search by text (including keyword or speech)
- Navigation with customized image categories

Some of the online image retrieval systems along with their advantages and disadvantages are described below:

2.2.1. Google Image Search

Google image search is one of the most known and used general-purpose image search engine. The success of Google image search engine is due to the link analysis algorithm implemented within it [34]. The fairly accurate image retrieval results is because of high efficient backend database that Google search engine uses and of the proprietary PageRank [57] [37] ranking algorithm. The degree of relevance between the images, their Webpages and the user's query is determined by using the link analysis technique described in the PageRank algorithm. Figure 2-4 shows the Google image search interface.



Figure 2-4 Google image search interface

There are some drawbacks of Google image search, though it is one of the most used image retrieval system. The drawbacks of this approach are:

- Search depends heavily on textual analysis (word occurrence for example), such as returning low precision result for general-term queries.
- Irrelevant results even for strictly defined queries.

Google image search is available online at <http://images.google.com>

2.2.2. PicSearch

PicSearch is a search engine dedicated solely for pictures only [34]. Picsearch uses a crawler agent to collect and index images in their local database instead of using a traditional text search engine. Picsearch claims it has a relevancy unrivalled on the web due to its patent-pending indexing algorithms. The indexed images are passed through advanced filters to eliminate offensive images.



Figure 2-5 PicSearch advance image search

Picsearch provides a very user friendly interface, designed to be simple. An advanced search option is offered too: users can specify the types of pictures (for example animations or faces), the colour (black and white or coloured), desired size (small, medium, large, wallpaper), and the orientation (portrait, landscape, square) of the pictures they are looking for. Figure 2-5 shows the PicSearch advance image search.

Detailed description of Picsearch is not possible because Picsearch algorithms are proprietary. Picsearch is available online at <http://www.picsearch.com>.

2.3. Feature Extraction

In image matching and retrieval task, feature extraction is a special form of reducing data dimensionality [86]. This approach is useful when the image sizes are large and reduced feature representation is required. Feature extraction process reduces the amount of resources required to describe the item (image). Analysis of images with extracted feature will allow more reliable modelling / learning / matching, etc. as irrelevant (noisy) information is suppressed. The performance of any identification algorithm is solely based on feature extraction technique. Extraction techniques that give less intra-class variance and large inter-class variance are regarded as good extraction techniques.

All extracted features could be coarsely classified into low-level and high-level features. Low-level features (such as colour, texture, shape and spatial layout) can be extracted directly from original images, whereas high-level feature (such as concepts and keywords) extraction can be based on low-level features [14]. Low-level features are used to find the similarities and dissimilarities between images. High-level features, on the other hand, cannot be extracted automatically and requires some training data or prior information. High-level semantics should be considered for better retrieval system, as low-level features might not always give satisfactory results. This makes high level features, application specific.

Feature extraction of the images in the training dataset is conducted off-line. This section introduces two features: texture and colour, which are used to extract the features of an image.

In this thesis, low-level features (e.g., texture, colour) are used to identify whether probing image is a duplicate or near duplicate of associated training images returned by text processing module. In the rest of this chapter, low-level features used in this thesis are briefly described. The textural feature used is Local Binary Patterns [65]. On the other hand, colour feature used is Colour Structure Descriptor [9].

2.3.1. Local Binary Patterns

Ojala and groups first proposed local Binary Patterns (LBP) in 2002 [65]. It has been found to be a powerful feature for texture classification. The most important property of the LBP operator in real-world applications is its robustness against the changes caused to monotonic gray-scale by illumination variations

[65][61]. Also, it make possible to analyse images in real-time setting due to its computational simplicity. There are different variant of LBP since its development such as Extended-LBP [65] [76], Improved- LBP [21], Multi Block- LBP [61] and Rotation invariant- LBP [65] etc.

LBP combines characteristics of statistical and structural texture analysis [65]. LBP as its name suggest is used to find the local patterns in the image. Every pixel is thresholded with the value of the centre pixel using 8-neighbourhood pixels. The basic LBP operates on a 3x3 kernel to encode the local spatial structure of image by comparing pixel intensity of the centre pixel with its eight neighbours. The pixels in this block are thresholded by its centre pixel value, multiplied by powers of two and then summed to obtain a label for the centre pixel. As the neighbourhood consists of 8 pixels, a total of $2^8 = 256$ different labels can be obtained depending on the relative gray values of the center and the pixels in the neighbourhood. An LBP code for a neighbourhood was produced by multiplying the thresholded values with the weights given to the corresponding pixels, and summing up the result as shown in Figure 2-6 [64]. Figure 2-6 also shows how average of the centre pixels neighbours (C) was derived. The average of the gray levels below the centre pixel is subtracted from that of the gray levels above (or equal to) the centre pixel.

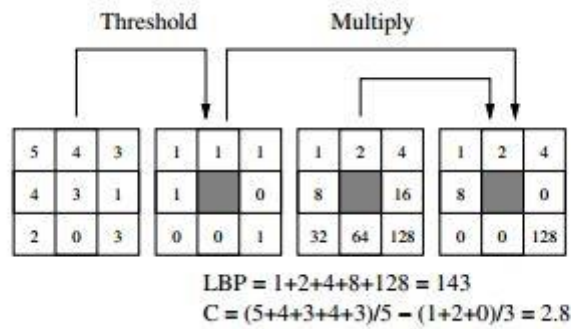


Figure 2-6 Calculating the original LBP code and a constant measure

Figure 2-7 shows an example of an LBP image and histogram [50].

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (2.1)$$

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$$

Where g_c and g_p denote the gray values of the central pixel and its neighbour, respectively, and p is the index of the neighbour. P is the number of the neighbours, and R is the radius of the circularly neighbouring set. Supposing that the coordinate g_c of is $(0, 0)$, the coordinate of each neighbouring pixel g_c is then determined according to its index p and parameter (P, R) as $(R \cos(2\pi p/P), (R \sin(2\pi p/P))$. The gray values of the neighbours not located at the image grids can be estimated by an interpolation operation. Three circularly symmetric neighbouring sets with different (P, R) are illustrated in Figure 2-8 [65].

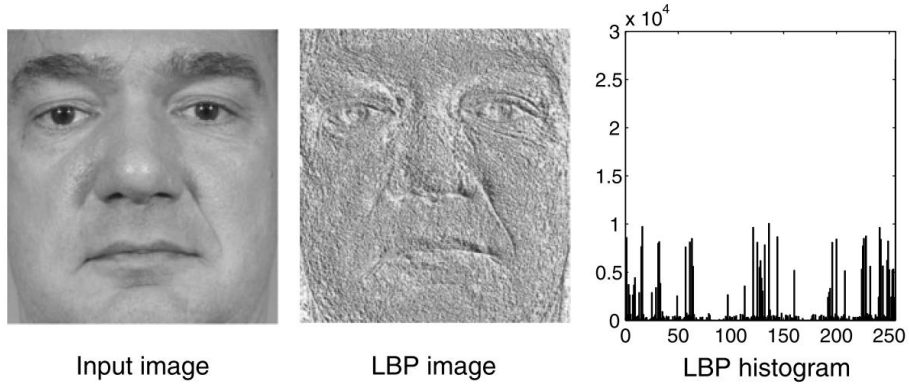


Figure 2-7 Example of LBP histogram

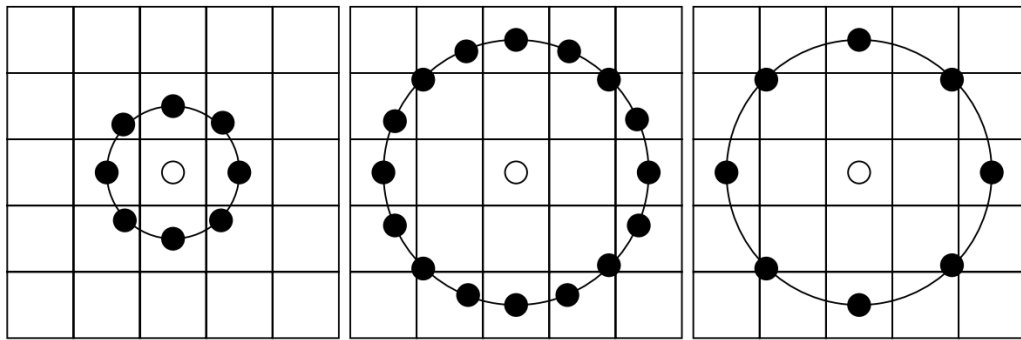


Figure 2-8 The circular (8,1), (16,2) and (8,2) neighbourhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the centre of a pixel

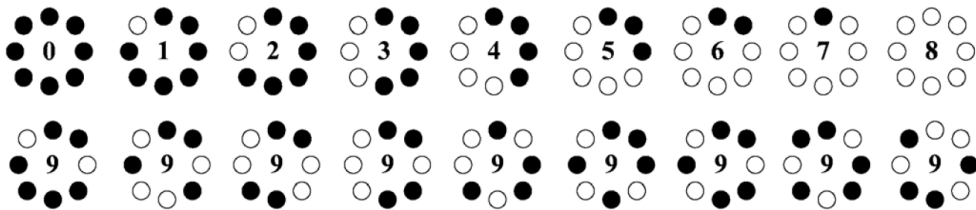


Figure 2-9 LBPs in a circularly symmetric neighbouring set of rotation invariant uniform local binary patterns

To obtain the uniform pattern, a uniformity measure is first defined as

$$\begin{aligned}
 U(\text{LBP}_{P,R}) = & |s(g_{P-1} - g_c) - s(g_0 - g_c)| \\
 & + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (2.3)
 \end{aligned}$$

Which corresponds to the number of spatial transitions (bitwise 0/1 changes) in the pattern. Based on the uniformity measure, the LBP descriptions of a texture image are defined as follows

$$\text{LBP}_{P,R}^{\text{riu}2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U(\text{LBP}_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \quad (2.4)$$

According to (2.4), LBPs with the U value up to 2 are defined as the uniform patterns, and its label corresponds to the number of “1” bit in the pattern. Non-uniform patterns are grouped into a category, labelled as $(P + 1)$. $\text{LBP}_{P,R}^{\text{riu}2}$ Can be calculated according to (2.4), and superscript “riu2” denotes rotation-invariant uniform patterns with $U \leq 2$. Hence, $\text{LBP}_{P,R}^{\text{riu}2}$ has independent $P + 2$ output values. For example, $\text{LBP}_{P,R}^{\text{riu}2}$ with values of (8,1) are shown in Figure 2-9 [65]. These uniform patterns represent the microstructures of an image, such as bright spot (0), flat area or dark spot (8), and edges of varying positive and negative curvature (1–7). The pixels in the non-uniform patterns are labelled as 9. After the LBP pattern of each pixel has been identified, a LBP histogram is calculated to represent the texture as follows

$$H(k) = \sum_{i=0}^W \sum_{j=1}^H f(\text{LBP}_{P,R}^{\text{riu}2}(i, j), k), k \in [0, K - 1] \quad (2.5)$$

$$f(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases} \quad (2.6)$$

Where K is the number of the patterns equal to $P + 2$ bins. The proportion of the pixels in the non-uniform patterns usually takes a small part in a texture image when accumulated into a histogram. Such strong capabilities of the uniform LBP feature to differentiate textures are based on the statistical properties of different patterns.

2.3.2. Colour Structure Descriptor

The Colour Structure Descriptor (CSD) [9] is one of the several colour descriptor defined in MPEG-7 for describing colour features of multimedia contents. CSD as its name suggests describes colour features, including both the colour of features and the structure of features. CSD detects the localized colour distribution of each colour in order to provide more precise and authentic description and basically it is based on colour histogram [35]. M quantized colour, c_m of colour structure histogram characterizes the CSD which is expressed as following:

$$h(m), m = 1, \dots, M \quad (2.7)$$

In expression (2.7) the value of $M \in \{256, 128, 64, 32\}$ and number of structuring elements containing one or more pixels with colour c_m is represented by bin value $h(m)$ [35]. It scans the image by an 8×8 structure element to express the local colour structure

in an image. The structuring element shows the functions like scanning image and counting the number of times particular colour is contained within it. The HMMD (Hue-Max-Min-Diff) colour space is used in this descriptor. Figure 2-10 shows double cone representation of HMMD colour space [10].

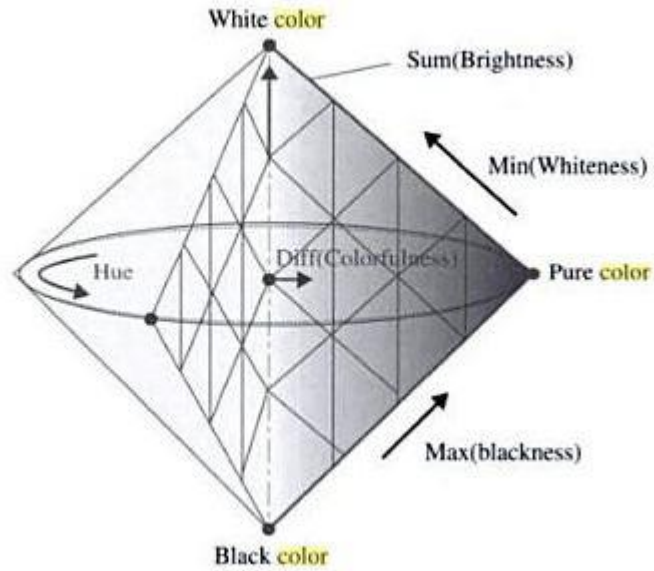


Figure 2-10 HMMD Colour Space

Table 2-2 shows the HMMD colour space quantization for CSD [9].

Table 2-2 HMMD Colour Space Quantization for CSD

Component	SubSpace	Number of quantization levels for different numbers of histogram bins			
		184	120	64	32
Hue	0	1	1	1	1
	1	8	4	4	4
	2	12	12	6	3
	3	12	12	4	2
	4	24			
Sum	0	8	8	8	8
	1	4	4	4	2
	2	4	4	4	4
	3	4	4	4	2
	4	2			

The sub sampling factor of p given the image size is given by following expression:

$$p = \max\{0, \text{round}(0.5 \log_2 WH - 8)\} \tag{2.8}$$

$$K = 2^p, \quad E = 8K$$

Where

- W, H image width and height, respectively;
- $E \times E$ spatial extent of the structuring element;
- K sub-sampling factor.

An 8×8 element with no sub-sampling is used for an image smaller than 256×256 pixels. The value of $p = 1, K = 2, \text{ and } E = 16$ for an image having size 640×480 . Denote I be the set of quantized colour index of an image and $S \in I$ be the set of quantized colour index existing inside the sub-image region covered by the structuring element. As the structuring element scans the image, the colour histogram bins are accumulated which could be expressed as following:

$$h(m) = h(m) + 1, \quad m \in S \quad (2.9)$$

Thus, the number of positions at which the structuring element contains c_m tends to determine the final value of $h(m)$ and the variation between CSDs is figured by using the L1 distance measure. The inclusion of spatial colour information helps CSD to provide more accurate similarity retrieval. As this representation is more closely related to the human perception it's more useful for indexing and retrieval. The high retrieval accuracy could be obtained from the Structure histogram due to its property of very well description of colour feature. Table 2-3 shows the performance of CSD for varying numbers of bins and bit quantization [9].

Table 2-3 ANMRR Results for CSD Using the HMMD Colour Space

# bins	8 bits	6 bits	4 bits	2 bits
184 bins	0.046	0.046	0.066	0.226
120 bins	0.049	0.051	0.067	0.230
64 bins	0.068	0.073	0.087	0.273
32 bins	0.105	0.107	0.130	0.342

3. IMPLEMENTATION

In previous section, we have gained some terminologies that were useful in designing our system competing in MSR-Bing Image retrieval Challenge. Also, we briefly explain different features that were used within our system. In this section we describe the overview of the system, its different components and its working procedure. CMuvis framework used for feature extraction in Bing Challenge for face bank and duplicate image detector is also described in this section.

3.1. System Overview

To provide an optimal solution for image search purpose, we would like to design an end-to-end system, which is able to assess the effectiveness of query term in describing the images crawled from web. The output of our system is a floating-point score for every image-query pair provided from MSR-Bing Image Retrieval Challenge at ICME 2014, which depict how efficient the query can be used to describe the image. Higher floating points indicate higher relevance and vice versa. The general idea of our system is presented in Figure 3-1.

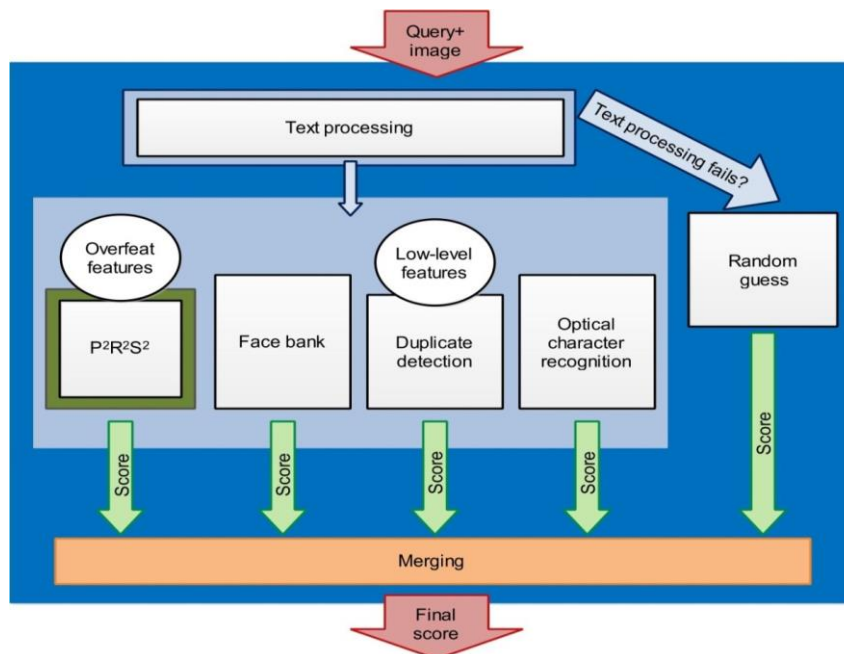


Figure 3-1 System Overview.

The system architecture consists of two main modules (Text processing, P²R²S²) and three complementary modules (face bank, duplicate image detector, optical

character recognition (OCR)). The input to our system is a data set table, which basically comprise of unique query-image pair combination. One image in the table may associate with more than one unique query and in similar manner one query may also associate with more than one unique image.

Text processing module is brain of our system, if it fails then every things else fails and output is solely based on random guess in decision-making. For each given image-query pair, text processing module takes query text as input and attempts to find the closest matching query from training dataset through pre-processing, query text clustering and query expansion operation. Training dataset is a dataset sampled from one-year click log of Microsoft Bing image search engine. Text processing module returns the image names together with number of clicks that are associated with the query in the training dataset. Overfeat features are extracted for the result images returned by text processing module. These overfeat features act as an input for $P^2R^2S^2$ module, which further transformed the feature vector. The output from $P^2R^2S^2$ is some relevance score for each image-query pair. The face bank come into play only if face is detected in input image and main aim is to enhance the query-image relevance. Low-level features Local Binary Patterns (LBP) and Colour Structure Descriptor (CSD) are extracted for the result images returned by text processing module and this feature act as an input to duplicate image detector. Duplicate image detector aim is to detect whether query image is a duplicate or near duplicate of associated training images return by text processing module. Optical character recognition searches for query words in images. The functionality of each module is presented below.

3.1.1. Text Processing

Text processing module has two modes (offline and online) to process data. In the offline mode, the module processes the training dataset to models from the query texts. During online mode, the module retrieves the “similar” training query texts for the probing query text using those models. In general, given a probing query text, text-processing module attempts to find the closest matching queries from the training dataset and returns the images associated with these query texts.

The query texts are single word or a sequence of two or more words arranged in lose grammatical forms with typographical errors (e.g. barrak Obama for Barack Obama). A large number of query texts contain information about geographical terms, person names and other identity names. Text processing module creates a set of word stems for each query text to give a unique semantic-ID. Aim of doing this is to merge several matching query into one entry. This operation include following steps:

- Removal of leading and trailing spaces.
- For each query text, we assume that it is Unicode encoded ASCII string. If it's not we skip the query text.

- We split the query text into words and start part-of-speech tagging for each word (mark each word as noun, verb, adjective, etc.). After tagging, nouns, verbs, adverbs and adjectives are kept and rest are discarded.
- Then we lemmatize the words using WordNet engine [16] so that their stems represent different forms of word.
- We also remove meaningless words for image retrieval. Our blacklist includes “image”, “picture”, “image of”, “picture of”, “pic”, “photo”, “free”, “wallpaper”, “background”, “printable” etc.

This module should find “Christmas pictures” when searched for “Xmas picture”. To accomplish better performance, we try to find all combinations of synonym of the probing query text when searched in training database. We use WordNet engine [16] to find all synonyms of a word. Query text clustering is done for each query text in training database. For example, “Christmas picture”, all occurrences such as “Xmas picture”, etc. is clustered as one entry. The clustering algorithm compares each query text to all the query entries within training database. Using 50 cores of Merope cluster [81], it takes less than 12 hours to complete the clustering of the entire training dataset.

If we are not able to find the exact query text and synonymous text matching for a probing query text, we rely on the query expansion algorithm to get more relevant images. The expanded queries can be either reliable or unreliable. Reliable queries are those queries whose semantic IDs are superset of the semantic ID of the probing query text. For example, “nba dunk contest”, and “2013 nba dunk contest” is reliable expansion of the probing query text “dunk contest”. If reliable queries are not found, we attempt to find unreliable queries. Unreliable queries are those queries whose semantic IDs partially overlap with the semantic ID of the probing query text. Unreliable query expansion results in a query that has different meaning than probing query text. For example, “man’s face” and “man’s face looking up” are unreliable expansion of the query text “man’s face looking down”.

If none of the above operations finds the relevant queries in the training dataset, Hunspell [80] spell checker is applied to correct possible typos. Autocorrected input query text goes through the same pre-processing, and query expansion algorithm to find the relevant queries in the training dataset. If even the auto correction fails, we assume that text processing failed and we resort to random guess in decision-making.

3.1.2. Features

Once text-processing module succeeds in returning the relevant images, two types of feature are extracted for the result images. Overfeat features are extracted which act as starting point for $P^2R^2S^2$ module as described in section 3.1.3. In the similar manner, low-level features are extracted for duplicate image detector module as described in section 3.1.5.

OverFeat: We use feature extractor named OverFeat [49] a CNN based image classifier and feature extractor for extracting features from dataset images. OverFeat was trained on the ImageNet dataset for classifying between 1000 image categories. The author provide small and large network with slightly different topologies. We use 1000-dimensional output layer of the smaller network as a descriptor for the dataset images, i.e. each image is described by its correlation with the ImageNet classes. Smaller network has an input resolution of 231*231 pixels. Thus, the input images are first resized to match input window for smaller network before extracting features from dataset images. To accomplish this, we first uniformly scale the image so that the smaller dimension equal 231, and then cropping the larger dimensions equally from both sides to match the required resolution. This way of resizing make sure that the aspect ratio will not distort, however some information on the image borders is lost. The method has produced satisfactory results in experimental testing.

Low-level features: Low-level features are used to identify whether the query image is a duplicate or near duplicate of training set images. We use CMUVIS framework to extract low-level features and after some experimental testing we decide on using Local Binary Patterns (LBP) [65] and Colour Structure Descriptor (CSD) [9] features for our purpose.

3.1.3. PCA-assisted Perceptron Regression with Random Subspace Selection ($P^2R^2S^2$)

PCA-assisted perceptron regression with random sub-space selection ($P^2R^2S^2$) module is core of our system. $P^2R^2S^2$ is a hashing system, which generates binary codes. It aims to encode high dimensional description of the visual features as compact binary strings. The advantage of binary coding is to retrieve and store big data efficiently as data is increasing rapidly every day. The general idea of $P^2R^2S^2$ is presented in Figure 3-2. Overfeat features extracted over training dataset act as original data for $P^2R^2S^2$. This feature are divided into sub-feature spaces via random subspace selection and for each subspace, separate codes are generated. Code generation is performed by PCA-assisted perceptron regression.

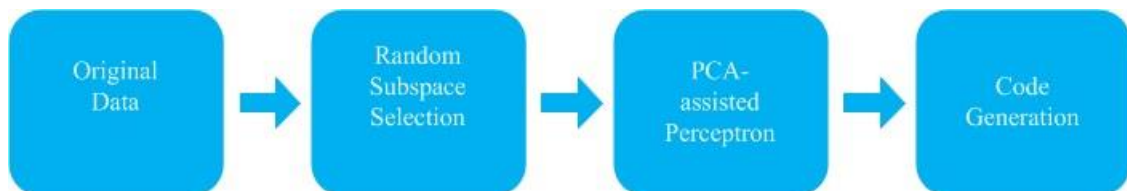


Figure 3-2 $P^2R^2S^2$ Flow Diagram.

In general, $P^2R^2S^2$ aim is to dig deep into the big data and reach to the information that is hidden among the huge number of samples and dimensions. The main idea is not to investigate data as a whole, but divide it into smaller subspaces

so as to reveal the hidden information. Division is applied on both feature space and data samples in a fully randomized way.

Traditional data mining technique tend to investigate further into dimensions of the sample space, identifying and selecting the most informative features, and finding out the correlation between different dimensions [25][40]. $P^2R^2S^2$ approaches this problem first by random selection among dimensions without replacement, forming different subspaces of the original feature spaces. The aim of random subspace selection is to improve stability, enable parallelization and decrease the memory cost. We form N sub-feature spaces each consisting of D randomly selected dimensions of the original feature space. N and D are set so that resulting feature spaces are overlapping and each dimension is used at least once. Unlike the traditional feature selection methods, sub-feature space formation in $P^2R^2S^2$ does not aim at decreasing the final feature space dimensions, but increase it, yet still keeping the investigation in smaller dimensions. Each randomly generated new feature space is later thoroughly examined using principal component analysis (PCA)-assisted perceptron regression to get binary code for each sub-feature spaces.

$P^2R^2S^2$ divide sample set into smaller partitions. We select randomly with replacement P partitions consisting of S samples each using bagging like approach. The aim of sample set partitioning is to reduce the amount of samples per examination over which parallelization is enabled so as to increase stability and accuracy of the applied learning method [36]. This operation increases the total number of samples to be examined, similar to previous step, but at the mean time decrease the number of samples per examination. Figure 3-3 shows simplified example of partitioning feature space and data samples [28].

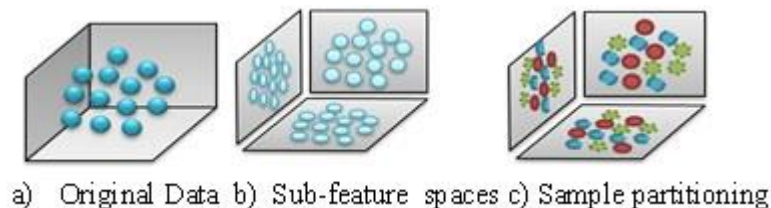


Figure 3-3 Examples of partitioning feature space and data samples

$P^2R^2S^2$ is aimed to generate reproducible and evocative representations. There are numerous ways for data representation, such as clustering, modelling, or code-word generation, etc. $P^2R^2S^2$ constructs data representation in such a way, which not only relies on a distance defined in the original feature space, but also investigates the sub-dimension relations of the given space. For each partition, we train a regressor to represent the behaviour of the samples in the corresponding partition. Regressors are trained in a supervised manner in contrast to clusters and models. The output must be presented for each sample used in the training of a regressor. Due to huge amount of different class labels, an output is set using unsupervised

data investigation. This aids for further investigation of the feature space, independent of the semantic relations indicated by the corresponding label.

Supervision in an unlabelled dataset is possible only due to assistance of PCA. The desired output of a given sample is obtained using the PCA projection of the corresponding sample vector [42]. We use the first V principal vectors to generate projections. If the projected value is greater than 0 (or the calculated average on that dimension), the corresponding desired output is set to 1 else it is -1. However these output values are just for initialization. The output values together with the distribution of samples at hand do not guarantee successful regressor training. In that case, an iterative approach is followed. The training samples are evaluated using the trained regressors and the corresponding responses for each sample are generated. The generated response is again thresholded using the initial threshold and new output values are generated that can be used in next iteration of training. In other words, if the response of a sample after training and propagation is greater than 0, the desired output corresponding to that sample is set to be 1 in the next training iteration. The mean squared error is computed after training and if it is lower than a predefined threshold value, training is assumed to converge. Each converged regressor is stored. An iterative approach of PCA-assisted perceptron regression over sample partitioning is shown in Figure 3-4.

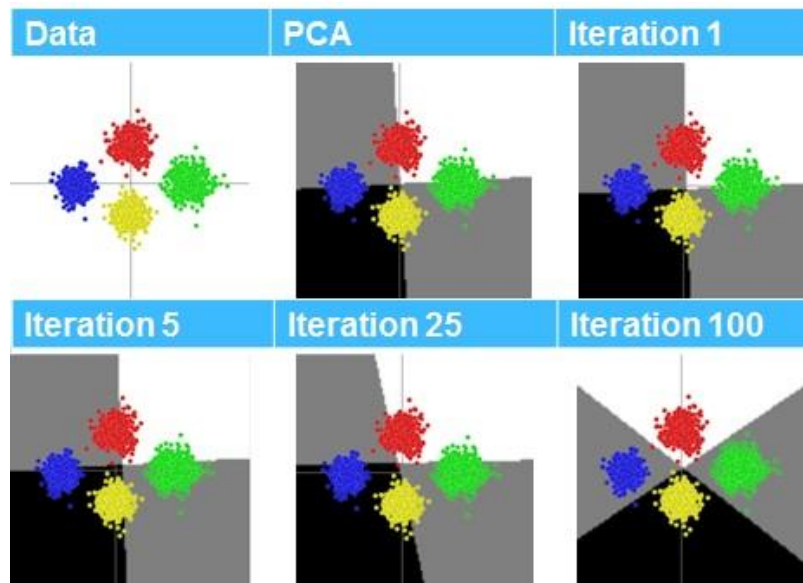


Figure 3-4 Examples of PCA-assisted perceptron regression over sample partitioning

A new sample vector is fetched to trained system, where its feature vector is divided into sub-vectors using the same randomly generated sub-feature spaces previously used in training. After that, each sub-vector is propagated through corresponding regressors and the responses are concatenated. The concatenation occurs also for different subspaces. In other words, a sample vector of 1000 is first divided into N different vectors of D dimensions, and then each D dimensional sub-vector is propagated through P regressors with outputs of V dimensions. So the

initial 1000 dimensional vector is transformed into an $N \times V \times P$ dimensional vector. Resulting vectors are used in retrieval, matching by a selected distance function.

In Bing Challenge, OverFeat features are extracted from all the training set images initially. During the test set evaluation, OverFeat features are extracted from the test image. Using $P^2R^2S^2$ module, we transformed the feature vectors of the test image and the set of associated training images returned by the text-processing module. L1 distance is used to compute the similarity between two feature vectors. The relevance score for each query-image pair is calculated by the following approach: First, the L1 distances between the query image and training images are calculated. Any examples with a click count lower than 2 are discarded, unless those are the only examples at hand. Then among those example vectors, the weighted average of the distances of the closest three vectors is calculated. The weights used in this calculation are obtained by natural logarithm of the click counts of the corresponding training images. Finally, this average distance is converted to a relevance score using a negative exponential function. The reliability score of the $P^2R^2S^2$ is determined based on the similarity of the query texts of the test image and the closest associated training images.

3.1.4. Face Bank

Face bank is complementary module, which aid in improving query-image relevance score, when face is detected in the query image. In Bing dataset a large number of query-image pairs consist of face images and thus extracting information from face images helps to improve retrieval rate. We created a face bank, which comprise of 2531 well-known celebrities from Posh24 [88]. For each celebrity, we collected 20 images with different facial pose angles to ensure a better recognition across all face angles. We train several MB-LBP based face detector to obtain pose-invariant face detection [66]. Detectors are trained for the following yaw-angles: $\theta = \{0^\circ, \pm 30^\circ, \pm 60^\circ\}$.

The output from highly confident detector is used to detector is used to obtain the final face localization. Further synthesized face feature vectors for given image was formed as relevance histogram in terms of celebrity list i.e., 2531 individuals. If a face is detected in an image, it is compared with every image in the face bank and a feature vector is formed as a histogram of relevant matches i.e. feature vectors have bins for all the 2531 individuals. Relevance of two faces is evaluated using the face recognition module provided in Intel Perceptual Computing [89]. This approach helps in finding the face similarities of images with respect to face bank celebrities. Figure 3-5 illustrates the synthesized face feature vector of barack Obama.

In the test evaluation of MSR-Bing Image Retrieval Challenge, for a given query text the pre-computed face feature vectors are loaded for the associated images

returned by the text-processing module and computed the Euclidean distances between the query images and associated training images. Less the distances between these images higher the reliability of face module.

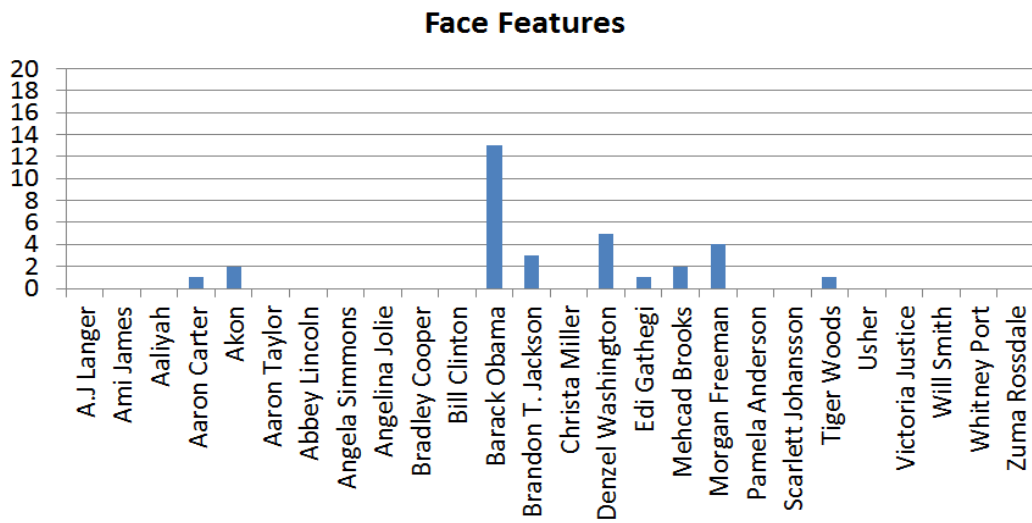


Figure 3-5 Synthesized face feature vector of Barack Obama

If a match is detected high relevance and reliability scores are returned. The final relevance score of our system is based on the face bank only when it is highly confident about positive match. The overview of the face bank module is shown in Figure 3-6.

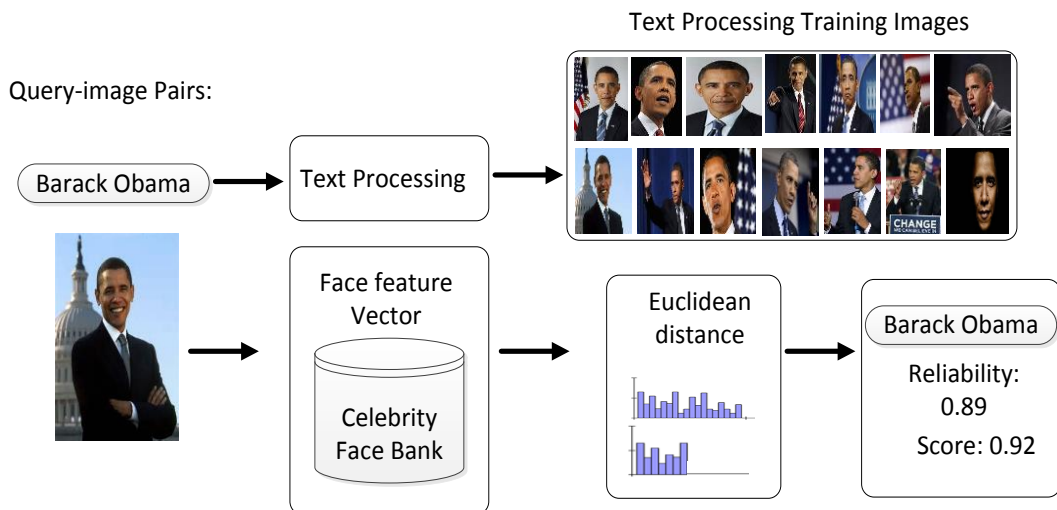


Figure 3-6 Overview of relevance evaluation using the face bank

3.1.5. Duplicate Image Detection

To enhance the query-image relevance result, we created a duplicate image detector. Its purpose was to identify whether query image is a duplicate or a near duplicate of a relevant image from the training dataset. The duplicate image detector used two low level features, LBP [65] and CSD [9].

First of all we extracted LBP and CSD features for the training dataset and given input image. To speed up the comparison process, the extracted features of the training set images were saved locally which were read later on. For comparing the relevance between the probing image and training dataset, Euclidean distance [87] was calculated for the input image and set of training images returned by the text-processing module. For both images to be relevant, we used a threshold distance Δ . The formula is given in equation below:

$$D_{LBP}(qimg, timg) \& D_{CSD}(qimg, timg) \leq \Delta \quad (3.1)$$

Where $D_{LBP}(qimg, timg)$ is the Euclidean distance of LBP features vectors extracted from the query image, $qimg$, and the training image, $timg$. Similarly $D_{CSD}(qimg, timg)$ is the Euclidean distance of CSD feature vectors. At first, duplicate images are searched only among the images associated with training query texts having the exact semantic ID and a click count higher than 1. If a duplicate among those images is detected, the module returns reliability score of 1 and the click count as the relevance score. If there are no more training queries which have same semantic ID as query text, then duplicate images are searched among the images associated with reliable query extensions. Both relevance and reliability scores are now set according to the Euclidean distance. In the final merging result, a tight reliability score threshold for using this module's relevance score is defined, to ensure that chances of considering false (near) duplicate images to be excellent matches are small. YUV and HSV features were also considered for duplicate image detector. However, individual and combined results were not better than combined result of LBP and CSD. Thus, we considered LBP and CSD as final features for duplicate image detector. The general idea of duplicate image detector architecture is presented in Figure 3-7.

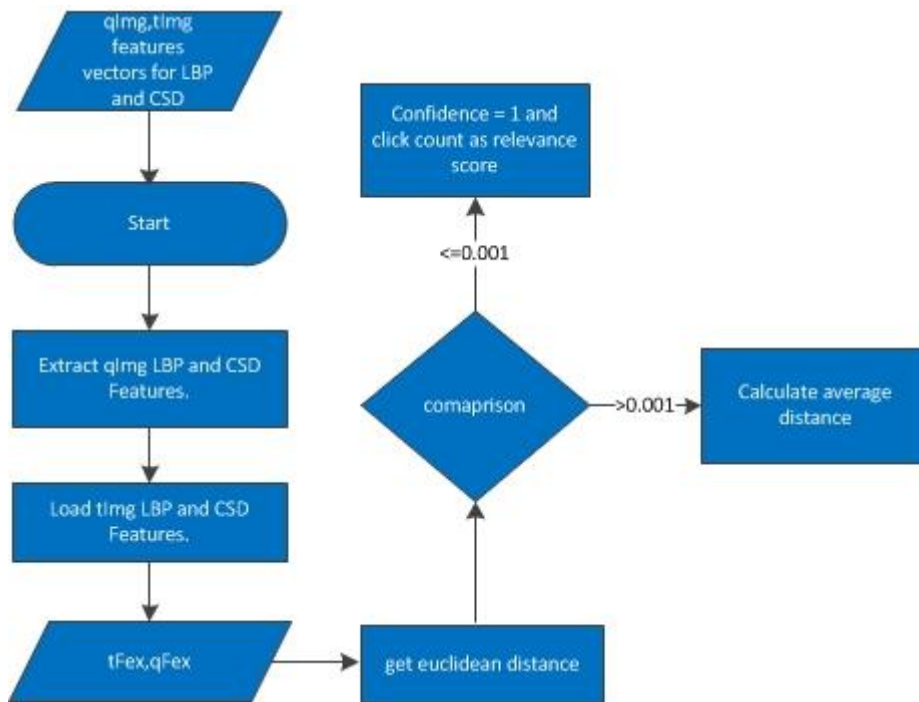


Figure 3-7 duplicate image detector architecture

Duplicate image detector extract LBP and CSD features from provided training set images in advance. Probing image and extracted features vector of relevant training set images returned by text processing module act as input for duplicate image detector. It extracts features (LBP, CSD) of query image and also loads the features vectors (LBP, CSD) of relevant training images. Query image features vectors (qFex) and relevant training images features vectors (tFex) act as input for Euclidean distance computing module. If the computed value ≤ 0.001 , confidence is set to 1 and click count as relevance score else average distance is calculated.

3.1.6. Optical Character Recognition

Optical Character Recognition (OCR) is used only when the probing query text is not similarly to any training set query. OCR is performed over the ranking images using Tesseract [55], which is a widely used OCR toolbox. All detected texts are compared with the probing query text and if the texts are overlapping, the OCR modules returns high relevance and reliability scores.

3.1.7. Merging

Each module gives a relevance score and a confidence score to a query image pair. The merging algorithm ensembles the results of all the modules to determine the final relevance score. Duplicate image detector, face bank, and OCR text module find out whether the probing query text and the probing image are highly relevant. If none of these modules is confident, the system uses the relevance score from the

$P^2R^2S^2$. Each module has its own relevance score range. The order of the range, from largest values to smallest values is shown in Figure 3-8.

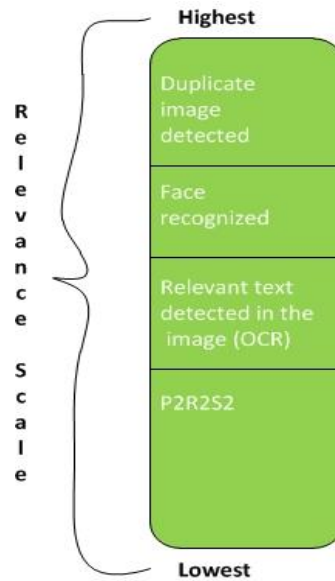


Figure 3-8 Relevance Scale Range

The OCR text module gives high relevance score if the probing image contains the query text. However, we assume that end users prefer more graphically appealing images to textual dominant images. Thus, we use the OCR text module only if the $P^2R^2S^2$ module confidence score is low. The threshold values used in the merging algorithm are empirically determined using the development dataset. Merging algorithm comprises of following steps:

1. Duplicate image detector relevance score is returned if duplicate image detector reliability score is greater than or equal to threshold_1.
2. Face bank relevance score is returned if face bank reliability score is greater than threshold_2 and face bank relevance score is greater than threshold_3.
3. OCR relevance score is returned if $P^2R^2S^2$ reliability score is less than equal to threshold_4 and OCR finds query text.
4. $P^2R^2S^2$ relevance score is returned if none of the above hold true.

3.1.8. Secondary Submissions and Other Considered Methods

We make three different submission i.e., master submission along with two secondary submission for the MSR-Bing challenge 2014. Every query-image pair is individually evaluated without comparing the test images associated with certain query text during master submission. For second submission, we have tried to exploit the relations of test images with each other. We assumed that for each given query-image set, if the image is relevant to query text, and then there are probably more images similar to that relevant image. Also to prohibit chances of having two similar irrelevant images in a query image set, irrelevant images are selected ran-

domly in a query image set. Using the assumptions stated above, the $P^2R^2S^2$ module compared the transformed feature vector of each test image in the test query image set with the feature vector of rest of the images in the test set. However, the module didn't compare it with feature vectors of the matching training set images as in master submission, which is already explained, in section 3.1.3. During second submission, no changes were made to face bank and duplicate image detector, only threshold value while merging is slightly changed. Also, OCR module was excluded during this submission. Over development dataset, the second submission was better than master submission. As this submission is based upon assumptions given above, it is not a general image retrieval solution due to which we decide not to use it as master submission despite the fact that it works better than master submission over development dataset. The third submission was almost similar to master submission but the only difference was we compared the overfeat features directly without transforming them using $P^2R^2S^2$.

For this challenge, we also considered to utilize Learning To Rank (LTR) over big data. The detail description of LTR is described in [17]. We adopted Rank-Boost (AdaBoost ranking) [33] and AdaRank [75] as two iterative ranking algorithms to produce a retrieval rank list over big data which sorts images according to their degree of relevance to the query. A decision stump performs as weak ranker, and the data was organized list wise in all cases. We selected 20 iterations for the algorithm to fulfil the discriminative power of our AdaRanker, while avoiding the extensive computation in further iterations. We used Normalized Discounted Cumulative Gain (NDCG) as an evaluation measure. The queries from the development dataset together with associated training images are used for training. The relevance for each query-image pair was determined based on the similarities between the query text, the training text and their click counts. The results obtained for MSR-Bing challenge 2014 using LTR were not as good as expected and due to this reason we didn't include this module in our proposed system.

3.2. CMuvis Framework

CMuvis framework is developed to process massive amount of data (BigData) efficiently. It is based on distributed computing model (client server), under service-oriented architecture. It provides a robust, fault-tolerant and power efficient processing of unstructured data, achieving high throughput and reducing processing time. This framework provides low latency, avoids duplication of data which is already "Big" and efficient access to information. In simple terms it's a divide and conquers technique where the high volumes of data are divided into smaller units called trunks each of which may be executed independently on any node in the cluster, a key to CMuvis framework. CMuvis framework support cross-platform i.e. it can run in Windows and Linux environments. The general idea of CMuvis framework is presented in Figure 3-9.

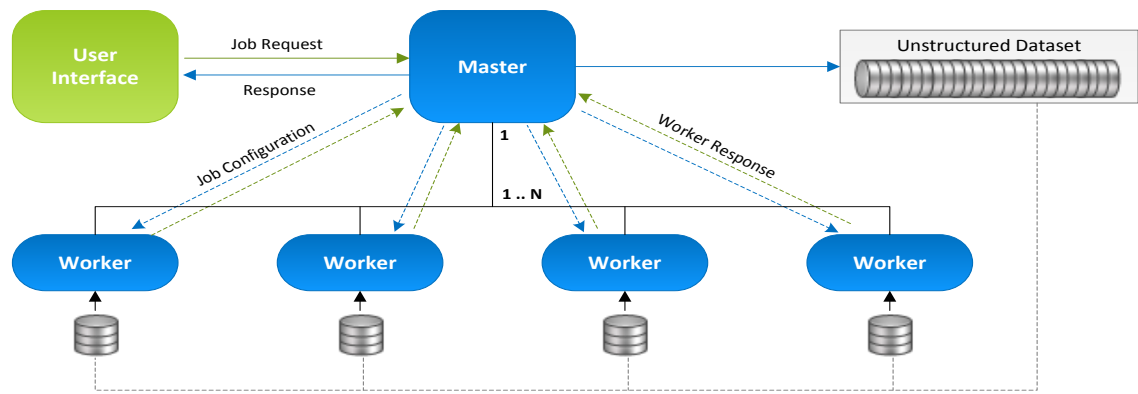


Figure 3-9 CMuvis System Overview

On receiving the task from user, master create dynamic configuration of the input task for workers and distribute the task to workers. The response from workers is merged by master to create a single output on user end. The different components of CMuvis Framework and its working procedure are described in this section.

3.2.1. Master

Master is responsible for interacting with the user interface and distributing tasks to workers. Master receives the task in XML format over HTTP protocol. After validating the input configuration, it is passed to the master controller.

Master controller is responsible for controlling the communication (Master-Worker and Master-User Interface). It also creates dynamic configuration of the input task for workers and the Data Partition module, it performs data partition if required. After creating configuration for the workers, the master controller passes that information to workers handler, where the worker processes are started and actively monitored. The number of workers, which are started, depends on the available computational resources. This section describe different component used within master.

Master Controller

End-to-End System communication is possible only due to master controller. This component controls the communication between master and worker i.e. master controller starts a server and waits for clients to connect. Master controller acts as a bridge between user interface and worker. After creating configuration of an input worker task, it forwards this information to worker handler.

Worker Handler

The information pass from master controller is received by worker handler component, where it create worker inventory and start the worker processes. Available computational resources decide how many worker processes to start simultaneously. The active processes are monitored under supervision of worker handler.

User Interface Controller

This component is responsible for controlling the communication between user interface and master i.e. UI controller start web server. Master is connected to UI controller as soon as task is forwarded from UI controller. The communication between master and UI controller is terminated when web server is closed.

3.2.2. Worker

Worker performs the job allocated by Master through an XML configuration. The master has a one-to-many relationship with the worker. At a time, there can be one or more workers executing depending on the type of task. For example, getting active database feature, only one worker is required. And, in case of extracting data features or querying, multiple workers are required for processing.

Each worker is assigned a designated data partition to work on. Workers have an active communication channel open with the master, and keep the master informed about worker status. It is worth mentioning here that a worker can perform the task without communicating with master independently as well. The general idea of worker architecture is presented in Figure 3-10.

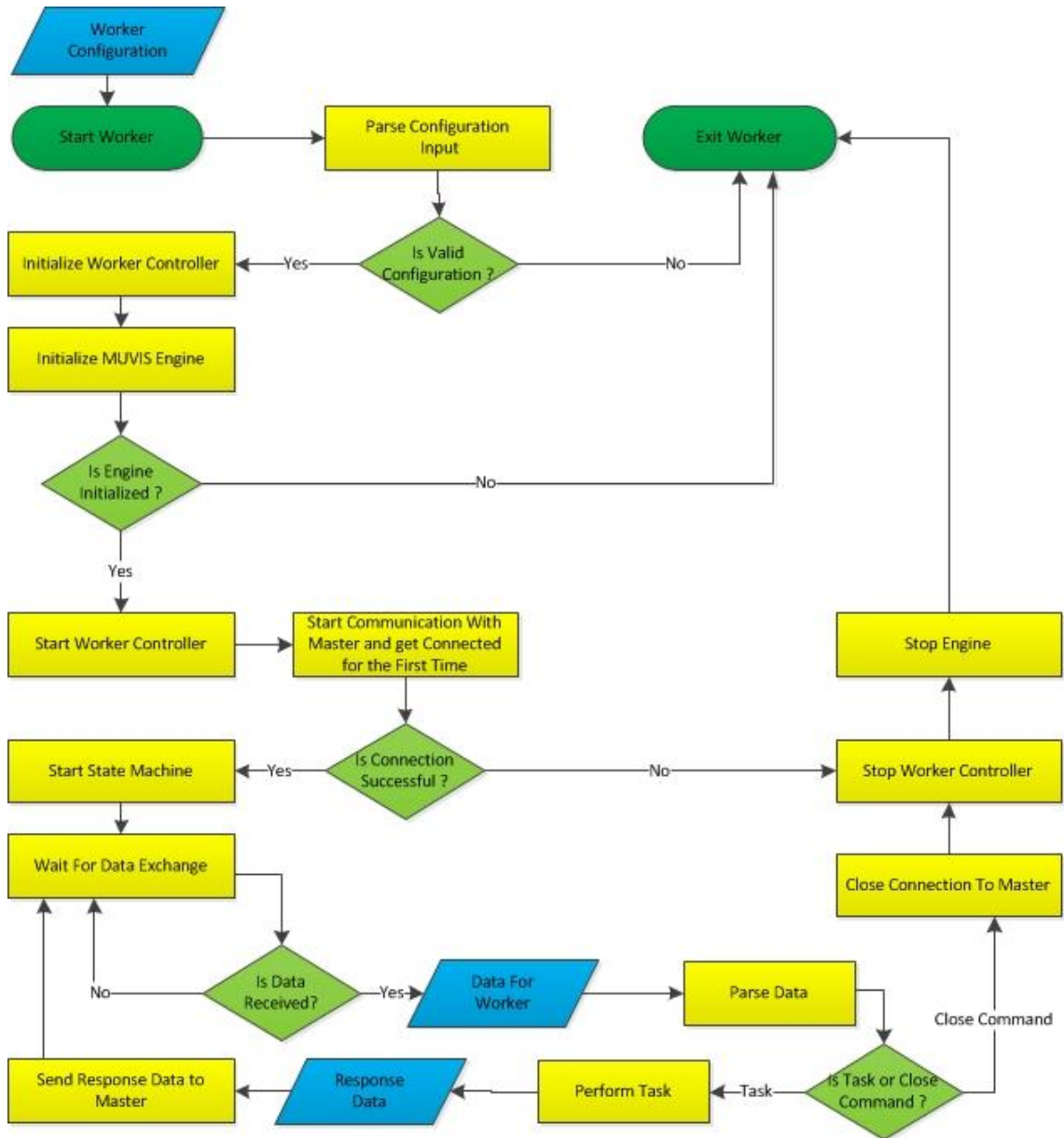


Figure 3-10 Worker architecture

User assigned the task to master in XML format, which is communicated over HTTP protocol. Master configured the task and divides it into smaller sub-problems based on number of trunks for give database. Smaller problem is passed to worker in XML format, which is communicated over Socket. Workers are reusable i.e. worker process task from master and after completion, worker wait for other incoming task unless CLOSE command is issued by master to close communication with the worker. At first, worker gets started whose objective is to initialize worker controller and start it so as to process the task given by master via master-worker communication. Once worker controller gets started, it initialize MUVIS engine, which load share library to hold generated DLLs by different modules. Worker connects to server and start communicating with master over socket. Once worker have an active channel open with master, asynchronous XML data is exchange between them. Worker then starts the state machine where that

actual parsing of the configured task is done and given to MUVIS engine. Master can check the status of worker state machine to figure out whether state machine is idle or running some task.

Worker sends the processed result back to master in XML format. Once, master receives result back from worker, it has two choices. It can either forward another task to worker or issue close command to terminate connection with worker.

Worker Controller

This is the brain of the worker and is responsible for running the task provided from master via worker state machine. The worker state machine takes one task at a time and it is re-usable. The task provided is in XML format and unique tasktype within task denotes the type of the operation to be performed. For example, create database operation, load database operation, append image operation, feature extract operation, etc. The communication with server started once MUVIS engine initialized whose objective is to load shared library to hold generated DLLs by different modules. The configured task from master is then forwarded to worker state machine where task is parsed and given to MUVIS engine. Finally, the result is send back to master in xml format.

Worker Sub-System

The general idea of worker sub system is presented in Figure 3-11.

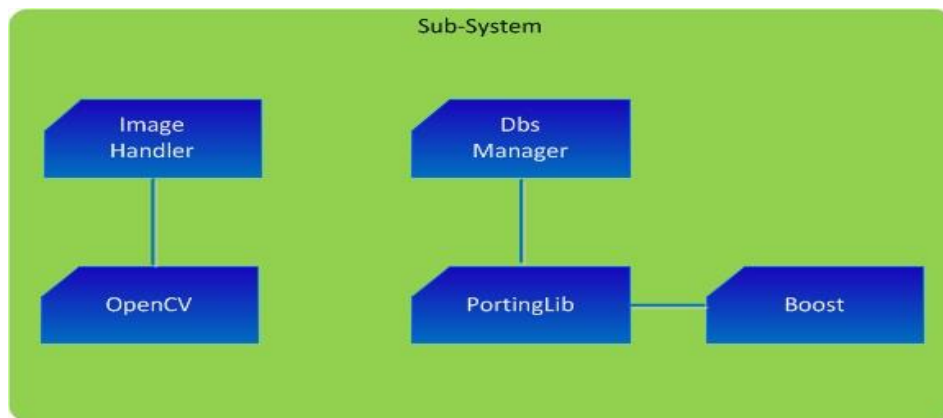


Figure 3-11 Worker Sub-System

Worker sub-system comprise of different module whose objective is to perform operation based on provided task from master. All the image related operations are handled through image handler module and this module is based on OpenCv library which is cross platform. Dbs Manager Module is backbone of our framework where different functionalities i.e. feature extraction (Removal), Consistency check; Hierarchical Cellular Tree (HCT) indexing [60] (removal), etc. are performed. Porting Lib module comprise of various useful components i.e. communication, xml parser, etc. Porting lib is based on Boost library, which is an open source based and is cross platform. Data sharing module is responsible for sharing data across several module in a single running processes.

3.2.3. Worker Library Dependencies

Worker module depends upon libraries generated from MTL, PORTINGLIB and DBSMANAGER modules. PORTINGLIB and MTL modules are static type while DBSMANAGER module is of dynamic type.

MTL

This module include the functionality that is needed to parse the feature xml during append or remove visual feature operation. The structure of feature xml is presented in Figure 3-12.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<muvisfeatures>
  <Indexing>NONE</indexing>
  <featurelist>
    <feature>
      <name>LBP</name>
      <featuretype>visF</featuretype>
      <noParam>2</noParam>
      <noSubFeatures>1</noSubFeatures>
      <subfeature>
        <param>7.000000</param>
        <param>5.000000</param>
      </subfeature>
      <action>ADD</action>
    </feature>
    <feature>
      <name>CSD</name>
      <featuretype>visF</featuretype>
      <noParam>1</noParam>
      <noSubFeatures>2</noSubFeatures>
      <subfeature>
        <param>32.000000</param>
      </subfeature>
      <subfeature>
        <param>256.000000</param>
      </subfeature>
      <action>ADD</action>
    </feature>
  </featurelist>
</muvisfeatures>
```

Figure 3-12 Sample Feature XML

The information within *muvisfeatures* tag is the parameters needed for appending feature. The value within *indexing* tag denotes whether to index the active database or not? The feature information within *feature* tag is feature that needs to be extracted. The *name* tag within *feature* tag denotes the name of feature that need to be append. The *featuretype* tag denotes the type of feature that needs to be extracted i.e. visual feature, aural feature, video features. The *noparam* tag denotes the no. of parameter for the feature that need to be appended. The *noSubFeatures* tag denotes the no. of *subfeatures* that need to be extracted for particular feature i.e.

RGB, YUV, etc. The information within *subfeature* denotes the parameter needed for appending features for active database. The *action* tag within feature tag denotes whether feature need to be add or remove from active database.

On receiving the feature xml stream from worker, MTL load it and start parsing the xml whose structure is presented above in Figure 3-12. Finally, the acquired values are stored in a container which is accessible to worker.

Portinglib

Portinglib module is a collection of numerous shareable helping components. These components are discussed below:

1) Communication

The task xml from user interface is passed to master via Http connection in a synchronous trend while configured task is forwarded from master to worker via socket in an asynchronous trend. Master receives the response xml back from worker over socket, which finally reach to user interface via http.

2) Xml Parser

Xml parser read an input stream and translates it to a data structure, which is manually inspected and retrieves the information that is needed. Also editing can be done on data structure and can be saved into xml file.

3) File Handler

File handler is responsible to manipulate files and directories within our system. Manipulation operation includes deleting file and directory, getting title of directory, getting number of file in directory, etc.

4) Others

Timers are used for measuring elapsed time, getting system time, starting timer, stopping timer, etc.

DbManager

This module is backbone of our CMUVIS framework. All the visual feature extraction (append/removal) related functionalities are encapsulated with in this module. Feature extraction (Fex) API functions provide necessary handshaking and information flow between CMUVIS application and feature extraction module. Each visual FeX module is implemented as Dynamic Link Library (DLL) with respect to Fex API. For example FeX_LBP, FeX_CSD, etc.

Hierarchical Cellular Tree (HCT) Indexing algorithm is functionality within DBSMANAGER which depending upon the available feature (visual/aural) and CMUVIS database item type (video/image) performs a complete reindexing scheme with the available indexing type (visual/aural/full). The objective of HCT indexing is to work with progressive query (PQ) in order to provide the earliest possible retrievals of the relevant items.

This module also provides functionality to check the consistency of CMUVIS database to make sure no feature files and indexing file are corrupted. It automatically corrects the corrupt files.

3.2.4. Worker Class Diagram

The following class diagram gives an overview of worker part of the system by showing classes and relationships among them. Each major class are described in next section: class responsibilities 3.2.5.

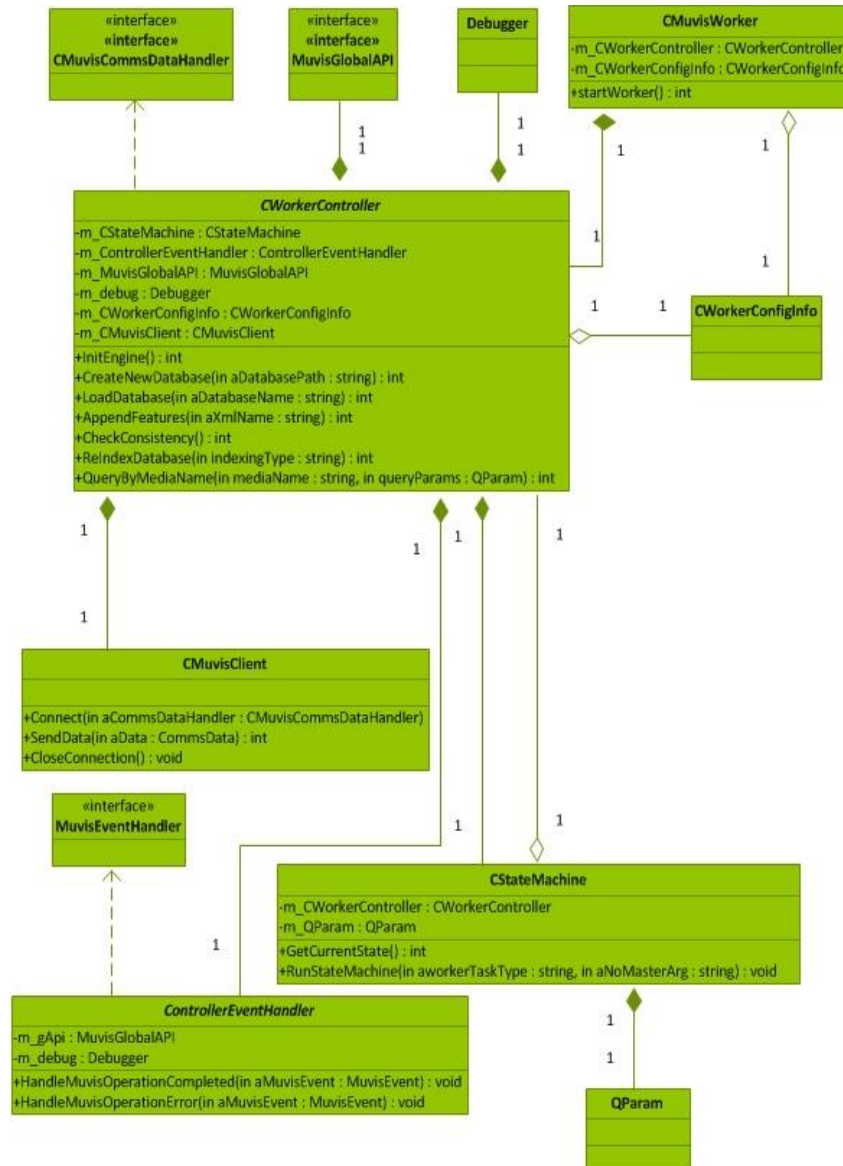


Figure 3-13 Class Diagram, Worker

3.2.5. Class Responsibility

The following table describes the responsibilities of some of the major classes contained in the class diagram of worker part of the system.

Table 3-1 class responsibility, CWorkerController

CWorkerController
<p><i>Responsible area:</i> It is brain of worker and is responsible for running the task provided from master via worker state machine.</p>

Table 3-2 class responsibility, CStateMachine

CStateMachine
<p><i>Responsible area:</i> It is responsible for executing global API function i.e., create database, append media item, etc.</p>

Table 3-3 class responsibility, CMuvisClient

CMuvisClient
<p><i>Responsible area:</i> It is responsible for handling client connection with server.</p>

Table 3-4 class responsibility, ControllerEventHandler

ControllerEventHandler
<p><i>Responsible area:</i> It is responsible for handling event raised on completion of worker controller operation (create database, load database, etc.).</p>

Table 3-5 class responsibility, CMuvisCommsDataHandler

CMuvisCommsDataHandler
<p><i>Responsible area:</i> It is responsible for handling event related to server and client communication.</p>

3.2.6. Sequence Diagram

The following sequence diagram describes the operation that is carried out by worker to perform the task forwarded by master via XML configuration. The steps are:

1. Master start worker and forward the task information. Worker then parses it and start worker controller.
2. Worker controller start muvis engine and establish asynchronous communication with master. Once communication started, worker controller start state machine where task is performed. The result is communicated back to master asynchronously.

3. Worker controller close state machine and worker close worker controller.
4. Finally worker gets close.

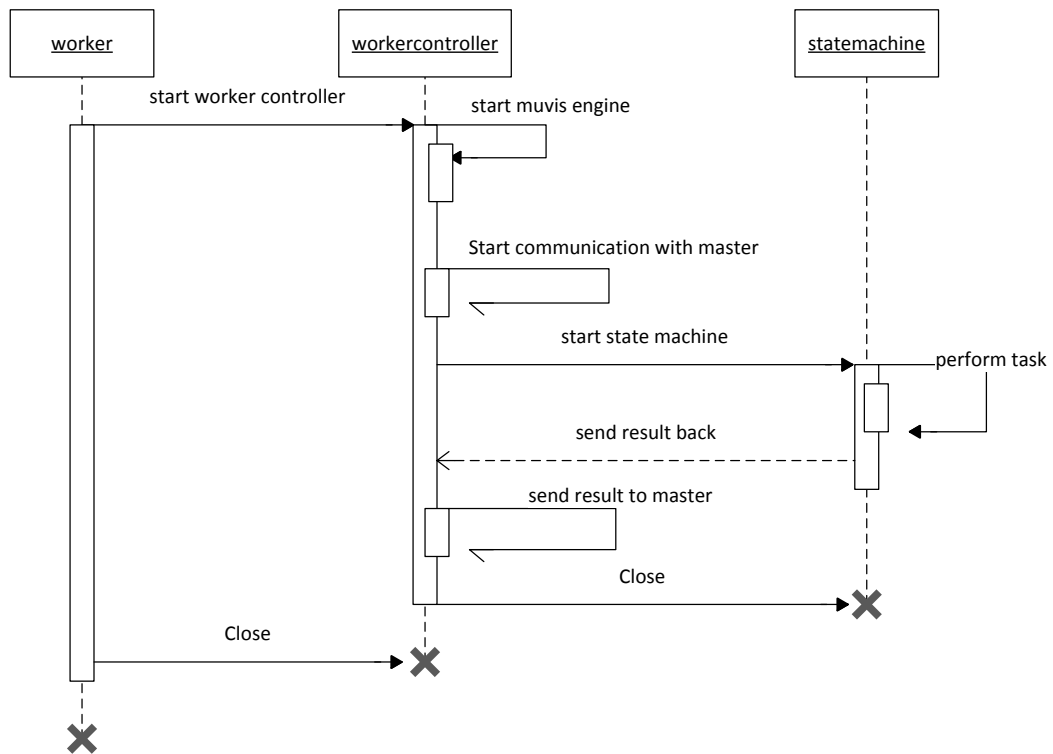


Figure 3-14 Sequence Diagram, Worker

3.3. Feature Extraction Framework

CMUVIS framework can support following types of multimedia databases:

- Audio/Video databases which is basically collection of audio/video clips.
- Image databases which is basically collection of still images
- Hybrid database which is collection of both audio/video databases and image databases.

Currently, CMUVIS framework support only image database and other will be implemented later on. CMUVIS supports visual feature extraction in such a way it allow third party to develop feature extraction module independently and integrate into CMUVIS without knowing the details of the CMUVIS applications. In the rest of the chapter, visual feature extraction framework will be explained in detail.

3.3.1. Visual Feature Extraction Framework

Image features are extracted directly from 3-byte RGB frame buffers obtained by decoding images. In order to query a media item (image) within CMUVIS database, the associated feature of media item need to extract and FeX Module carries out this extraction. These FeX modules can be implemented independently and then integrated into CMUVIS framework dynamically (during run-time) via a specific Feature Extraction Interface (FeX API). The detail description of Fex framework

can be found in [47]. The Fex module interaction with CMUVIS application is shown in Figure 3-15.

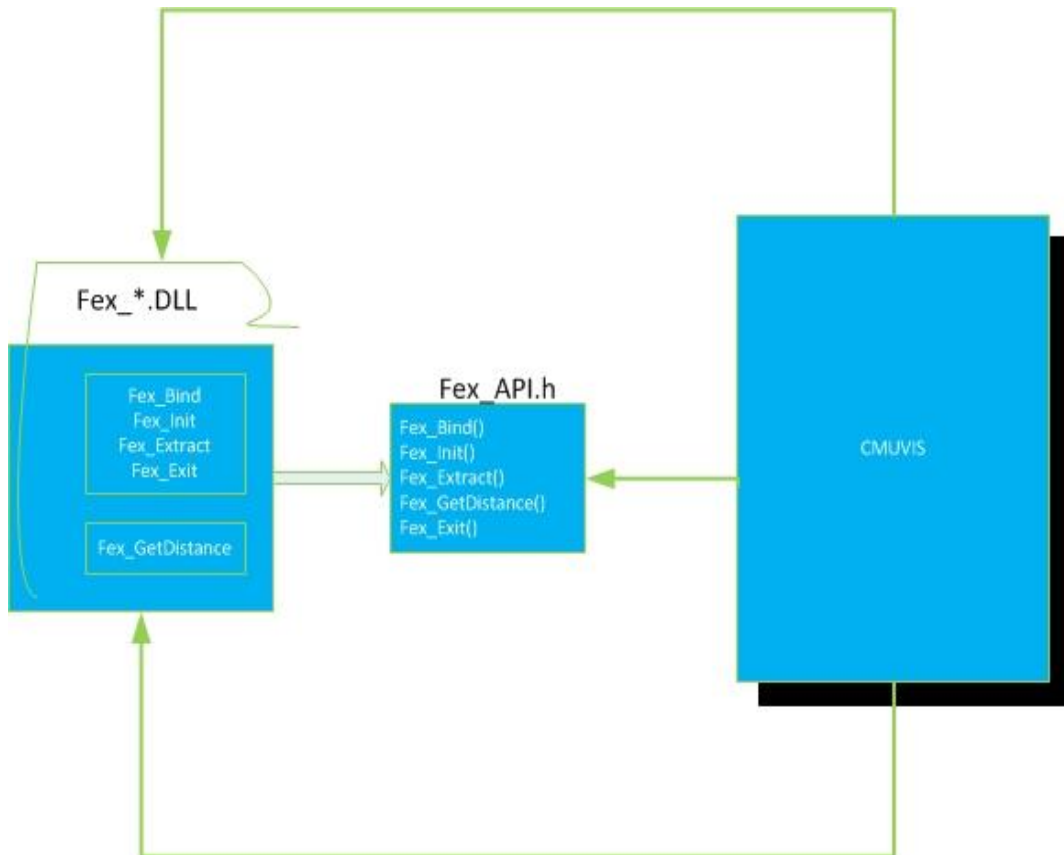


Figure 3-15 Fex module interaction with CMUVIS application

Whenever master starts worker, it checks for the FeX modules (DLLs) in FeX module path (**CMUVIS_FEX_MODULES_PATH**). If this environmental variable is not set, it will search within current program path. It detects and links all the FeX modules functions available under FeX module path. Next, Fex_Bind function is called to establish handshaking operation between CMUVIS framework and a FeX module. During this operation, associated FexParam structures are filled by the modules value i.e. feature name, fourcc code of feature, number of parameter of feature, feature parameter default value, frame type, etc. Once binding of FeX module is done, worker calls Fex_Init function, which uses the FeX parameter provided by user to initialize the FeX module. For each image in the database, worker calls the Fex_Extract function with the FrameParam structure and function returns the feature vector along with the vector size. Once all the sub-features with certain parameters are extracted, fex_Exit function is called to reset the FeX module with FexParam structure that is already filled via Fex_Bind function.

4. EXPERIMENTAL RESULTS

In this chapter, we have evaluated the performance of our proposed image retrieval system from Bing user click log for MSR-Bing challenge 2014 and two different datasets are used for the evaluation process. First evaluation is based on development dataset and second evaluation is based on test dataset. This chapter also includes the detail description of two datasets used.

4.1. Dataset

The detail description of the training dataset can be found in [71]. The dataset was sampled from one-year click logs of a commercial image search engine. It consists of a big table with 23.1 million triads. A triad $\langle K, Q, C \rangle$ means that the image “K” was clicked “C” times in the search results of query “Q” in one year (maybe by different users at different times). Image K is represented by a unique “key” which is a hash code generated from the image URL. Similarly query “Q” is a textual word or phrase, and query count “C” is an integer, which is no less than one. One image may correspond to one or more entries in the table. One query may also appear in multiple entries that are associated with different images. There are 1M unique (in terms of URLs) image keys, that is, images in the dataset. Compared with the scalable image recognition works in existing datasets with hundreds to thousands of labels [4][32][51], training dataset has millions of open label set.

A users often click on one or more images that are relevant to the queries while doing image search, so most of the times Q in the triad is relevant to the image K, and the bigger the click count C is, image K is more likely to be relevant to the query Q. Image thumbnails are provided in a separated data file. The maximum height and width of an image thumbnail is 300, if the original image’s height or width is no less than 300. Otherwise, the original size is kept. For extremely tall or wider images (for example: 8000x100) we try to keep the size (i.e. width \times height) of the thumbnail as close as possible to 300x300 pixels.

For convenience, we call Q a “clicked query” of Image K, and K a “clicked image” of query Q, and call $\langle K, Q \rangle$ a “clicked image-query pair”, and the triad $\langle K, Q, C \rangle$ as “click data”. We also call the “clicked queries” of an image as the “labels” of the image. A unique $\langle K, Q \rangle$ pair only appears once in the table, which signifies that the click counts of a clicked image-query pair have been aggregated into one entry in the table.

4.2. Dataset Properties

The click based dataset has some unique characteristics in comparison to human labelled dataset. Compared to existing datasets [4][32][51], Clickture was constructed by a totally different approach by using search engine click logs. Clickture can potentially grow fast in both the amount of data that we can collect and its semantic coverage as it is by-product of commercial search engines. Likewise the potential growth could also be faster due to the increase of the image indexed as well as the number of users that are using image search. It also well reflects common users' searching and consuming interests as it covers the semantics (textual queries) that people desire to search and recognize in daily life. Labels in the Clickture are more accurate than the datasets that are using search results from search engines directly or user-input tags in social media portals, though they are less accurate than the datasets created by manual labelling [71].

4.2.1. Properties of Clicked Queries (Labels)

The properties of clicked queries are described as follows:

- In general, clicked queries are relevant to corresponding images but noises can also be observed. While doing image search, a user may mistakenly clicked some images not due to the reason that they are relevant to image that user is interested but due to the fact that it attract users' attention (for example, very unique or strange images). Clicked queries may also contain typos, for example, "Barak Obama" in Figure 4-1 [70]. The reason for this is due to the fact that search engines have a mechanism to deal with "typos", i.e., typos in a query will be recognized and the images related to the (automatically) corrected query will also be shown to users. Due to this, clicked images may be associated with queries with typos. The click count is a good indicator of confidence of the clicked query to the image. Thus, grouping multiple visually duplicate and/or similar images into one cluster can reduce noises in clicked queries.
- The dataset is constructed from users search and click log. Due to this, the clicked query list is not a complete description of the corresponding image (though there is not any existing image dataset that attempt for a complete description of all semantics in the pictures in the dataset). It is for sure that during search, clicked queries do include labels that users are interested and these labels have good "representativeness". Although, these labels are not complete.
- While doing searching, most of the users search for specific things, for example, an actor's name ("Tom Hanks"), a product using brand and model ("Toyota Prius V"), a specific event ("Oscar 2013"), or an attraction ("Lake Washington"). Large numbers of specific queries in the dataset are

observed. Though, general queries like “cat”, “dog” and “sunset” are also very common.

- Similar and near-duplicate clicked queries for an image are not merged or removed. Intention is to provide click data in the dataset for image retrieval. Quite a few similar clicked queries are observed for “popular” images (images with a large number of clicked queries). For example, “cat”, “cats”, “cat picture”, “image of cats”, “image of cat”, “cats images”, “kitten”, “image of kitten”, etc., may be the clicked queries of a same image. The reason for not merging similar queries for same image is to encourage to do some pre-processing of query text, for example, to remove “picture”, “image”, “image of”, and “picture of” from the labels.
- Some images have a large number of clicked queries, while on the other hand some only have one. Around 120K images in the dataset have 50 or more clicked queries, while around 220K images only have one label.




		
fall :113;fall pictures :85;fall leaves :48;fall backgrounds :33;fall images :28;fall foliage :21;fall colors :18;fall pics :16;fall trees :14;autumn images :13	barack obama :414;barak obama :60;barack obama pictures :44;barrack obama :21;presidents :12;pictures of barack obama :3;pictures of barak obama :2;images of barak obama :2;barrak obama :1;barack obama image :1	food :513;food pictures :13;pictures of food :11;food pics :5;picture of a food :4;fast food :3;food images :3;restaurant food :2;foood :2;food picture :2

Figure 4-1 Examples of clicked queries with click counts

4.2.2. Properties of Clicked Images

Given probing query, do all the images associated with that query in the dataset covers all or most of the semantic and visual variances? For example, will images with label “Apple” cover all of the variations below: apple as a fruit (with different colours and types), Apple as company logos (different colours, sizes, and backgrounds), and Apple company’s products (iMac, iPhone, iPad, iPod and their accessories)? Are images sufficiently representative for each case? Answer to these questions cannot be given right away, but can still argue that the images in the dataset are those that attract users’ interests and have large visual variances.

In dataset, some queries are associated large number of the clicked images, while some only with few. Actually, a large number of the queries (69%) only have one clicked images. Luckily, more than 240K queries have 10 or more clicked images. We can easily aggregate all the queries with their query counts on a same image key. Please note the “key” is generated from image URL, instead of

image content. So we cannot use the key to find all duplicates of an image. By comparing the thumbnails or MD5 hashes of the thumbnails, we can find “exactly” duplicate images. Using content-based signatures, the near duplicate images could be identified. Duplicates and near duplicate images can be used to improve the accuracy of the labels. An example of clicked images is shown in Figure 4-2.



Figure 4-2 Examples of Clicked Images

4.3. Construction of the Evaluation Datasets

The evaluation datasets are created to have consistent query distribution, judgement guidelines and quality. The construction of evaluation datasets includes the following steps:

- Random selection of sample queries, out of one-year click log of Bing search engine.
- Manual judging for large set of plausible image results is done for each query in order to ensure the high data quality and consistency.
- The results that are judged as detrimental. (e.g., adult or geopolitically inappropriate contents) are not considered.
- The queries with all results having identical labels (e.g., all Excellent) are not considered.

Dev Dataset File Format

The format of devsetlabel and devsetimage are:

DevSetLabel:

query <tab> image ID <tab> judgement (Excellent/Good/Bad)

DevSetImage:

image ID <tab> base64 encoded JPEG image thumbnail is processed so that larger dimension of width and height is at most 300 pixels.

4.4. Construction of the Training Dataset

The training dataset is a uniform random sample of one-year user click log of Bing search engine. The construction of training dataset includes following steps:

- All the clicked query-image pairs from one-year click log of Bing search engine in EN-US market are selected. A pair has N presence in the set, if it has been clicked N times within that year.
- Uniform sampling of the set is done randomly so as to retain exactly one million unique images based on following considerations: (a) be sufficiently large to represent real web-scale search and to train models, (b) be small in size to facilitate downloading and local development with available computing resources.
- For each image, all the clicked queries within that year along with corresponding click counts are outputted.
- The dataset is processed in order to minimize inappropriate image contents.

Training Dataset File Format

The format of trainclicklog and trainimageset are:

TrainClickLog:

image ID <tab> query <tab> click count

TrainImageSet:

image ID <tab> base64 encoded JPEG image thumbnail is processed so that larger dimension of width and height is at most 300 pixels.

Table 4-1 Dataset Statistics

	Distinct Queries	Distinct unigrams
Training Dataset	11,701,890	7,174,869
Dev. Dataset	1,000	4,144

4.5. Evaluation

Random sampling of training dataset is done to retain 100k images along with their click counts which are used to train P²R²S² module. The reason for using only 100k images out of training dataset is due to limitation of system. The system is tested by using development dataset which contains 80k query-image pairs, 1000 queries and almost 80k images. The performances of our methods are evaluated by using DCG_p which measures the ranking results against manually labelled development set. DCG_p For each query is calculated as

$$DCG_p = \alpha_p \sum_{k=0}^p \frac{2^{rel_k} - 1}{\log_2(k + 1)}$$

Where $p = \min(25, \text{number of images})$, $\alpha_p = 0.01757$, rel_k is graded relevance score of the result in position k in the labeled dataset, and $\text{rel}_k = \{\text{Excellent} = 3, \text{Good} = 2, \text{Bad} = 0\}$. The final evaluation is computed as the average of DCG scores for all the queries.

The evaluation method is changed from online to offline this year so as to encourage more teams to participate in the grand challenge. The teams are asked to download one compressed file (evaluation set), which contains two files in textual formats. Among the two, one is a list of key-query pairs, and the other is a list of key-image pairs. One full day (24 hours) is given to do predictions on all query-image pairs in the evaluation set. Also, the number of query-image pairs had been significantly increased this time as compared to previous one.

4.6. Partial Results on Individual Modules

There were 24 queries in development dataset and 4 queries in training dataset for which no matching query were found in training dataset, even with the text processing.

The DCG score was computed only over those queries within development dataset where face bank module is able to detect faces in query image and matching training images. In this case, the average DCG for random guess was 0.59961, while DCG with face bank module was 0.6611. There were 1894 cases within development dataset where positive face match was detected. The total number of images paired with these queries was 3103 and the rest of the query-image pairs were evaluated randomly.

The same testing as face bank module was used for duplicate image detector. Out of 42306 query-image pairs considered, the output of duplicate image detector was used only when a duplicate image was detected (3484 cases). In this case, the average DCG for random guess was 0.6533, while DCG with the duplicate image detector was 0.6859.

For OCR module, we conducted the similar testing. OCR was used for 143/8654 query-image pairs. In this case, the average DCG for random guess was 0.3820, while DCG with OCR was 0.3881. Though the result with OCR is only within the random score variance (0.0077), but we think that this module will perform better for test data.

4.7. Final Parameter Setting

The system parameters that we set for our master submission are shown in Table 4-2.

Table 4-2 Parameters values used in our master submission

Param.	Explanation	Value
Δ	Similarity threshold for duplicate images	0.001
N	Number of sub-feature spaces	100
D	Sub-Feature Space dimension	25
P	Number of partitions/regressors	20
S	Number of samples per partition	7500
V	Number of principal vectors used	20

4.8. Overall Results

DCG score for different version of our system over development and test sets are shown in Table 4-3. There is no logic behind Random score; it's just simple random guess. Master, Sub2, and Sub3 results are obtained using our master, second and third submission as explained earlier. To obtain OverFeat results, we set the relevance score according to the L1 distance of the OverFeat features of the probing query image and closest matching image from training set. Similarly to obtain OverFeat2 results, we set the relevance score according to L1 distance of the OverFeat features of the probing query image and other test images connected to the same query text. OverFeat2 is similar to second submission; the only difference is that it does not use $P^2R^2S^2$ to transform its feature. $P^2R^2S^2$ and $P^2R^2S^{2*}$ are similar to master and second submission, the only difference is face bank, duplicate image detector and OCR are not used. To obtain PCA results, we replace $P^2R^2S^2$ with principal component analysis. Looking at additional results on test set, it is quite clear that our main module could have performed well even without complementary module.

Table 4-3 DCG scores for different versions of our system over the development and test sets

	Random	Master	Sub2	Sub3
Dev. set	0.4704	0.5099	0.5361	0.5006
Test set	0.4858	0.5116	0.5463	0.5044
OverFeat	OverFeat2	$P^2R^2S^2$	$P^2R^2S^{2*}$	PCA
0.4974	0.5287	0.5082	0.5359	0.4945
0.5037	0.5406	0.5123	0.5473	0.5042

The number of query-image pairs utilized by each module when deciding the final relevance score is shown in Table 4-4.

Table 4-4 Number of query-image pairs where each module was used when setting the final score

Dataset	$P^2R^2S^2$	Face bank	DID	OCR	Random
Dev.	75080	650	3942	142	112
Test	316338	1158	2519	162	1040

5. CONCLUSION AND FUTURE WORKS

This section summarizes the thesis with concluding remarks. This section also discusses possibilities for improvements in the future.

5.1. Conclusion

This work showed a system to find the relevance of query text in describing the images crawled from web for image search purpose. This system contains two main modules (text processing and $P^2R^2S^2$) to find the relevance between query text and image list. To improve the evaluation accuracy of system three complementary modules are introduced. These complementary modules are:

1. Face bank
2. Duplicate image detector
3. Optical character recognition

This work is focused on extracting features for duplicate image detector module using CMUVIS frame work. The duplicate image detector is used to identify whether query image is a duplicate or near duplicate of a relevant image from training set on the basis of:

1. Local binary pattern (LBP)
2. Colour structure descriptor (CSD)

The idea of having complementary modules along with main $P^2R^2S^2$ is to improve the evaluation accuracy. The results showed that $P^2R^2S^2$ could have perform better alone on evaluation set (see section 4.8 Table 4-3). The reason is no learning is involved in complementary modules. However, the results showed improvement in evaluation accuracy for development set case. The system reached the value of 0.5099 and 0.5082 in terms of DCG_{25} on the development set with and without complementary modules, respectively.

5.2. Future Work

It is worth mentioning that text-processing module is brain of our system and output from text processing module act as input for all other modules. There is a very little machine learning involved in current text processing module. We may start from following direction in future for text processing module.

1. Learn more descriptive text feature. Some feature encoding methods can be considered.

2. Learn better representative model that combines both text and image features.

We can try to use smarter way to form different subspaces of original feature spaces instead of random selection in $P^2R^2S^2$ module. The boost in evaluation accuracy using Face bank was noticeably small. In this direction, the obvious possibilities of improvement are:

1. Collect more celebrities to create bigger face bank.
2. Utilize better feature extraction technique for representation of facial features.
3. Utilize deep learning in face bank.

For detecting duplicate image we can try using features other than LBP [65] and CSD [9].

REFERENCES

- [1] A. B. -Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis Metric from Equivalence Constraints," *Journal of Machine Learning Research*, No. 6, Jun 2005, pp. 937-965.
- [2] A. C. She and T. S. Huang, "Segmentation of road scene using color and fractal based texture classification," in *Proc. ICIP*, Austin, Nov 1994.
- [3] A. Gray, "The Intuitive Idea of Distance on a Surface," in *Modern Differential Geometry of Curves and Surfaces with Mathematica*, 2nd Edition, Boca Raton, FL: CRC Press, 1997, pp. 341-345.
- [4] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS* 2012.
- [5] A. M. Riad, H.K. Elminir, and S. Abd-Elghany, "A Literature Review of Image Retrieval based on Semantic Concept," *International Journal of Computer Applications (0975 – 8887)* Vol. 40, No. 11, Feb 2012, pp. 12-18.
- [6] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain "Content-based image retrieval at the end of the early years," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12, 2000, pp: 1349–1380.
- [7] B. Dinakaran¹, J. Annapurna, and Ch. A. Kumar, "Interactive Image Retrieval Using Text and Image Content," *Journal of Cybernetics and Information Technologies*, Vol. 10, No. 3, 2010, pp. 20-30.
- [8] B. Singh, W. Ahmad, "Content Based Image Retrieval: A Review Paper," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, Vol. 3, Issue. 5, May 2014, pp. 769-775.
- [9] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, and A. Yamada, "Color and texture descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, Vol. 11, No. 6, Jun 2001, pp. 703-715.
- [10] B. S. Manjunath, P. Salembier, and T. Sikora, "Introduction to MPEG-7: Multimedia Content Description Interface," *John Wiley & Sons, Inc. New York, NY, USA*, 2002, pages 396.
- [11] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-based image querying," in *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries*, in Conjunction with IEEE CVPR'97, 1997.
- [12] D. Feng, W. C. Siu, and H. J. Zhang, "Multimedia Information Retrieval and Management," *Technological Fundamentals and Applications*, 2003, pages 476.
- [13] D. Hoiem, R. Sukthankar, H. Schneiderman, and L. Huston, "Object-based image retrieval using the statistical structure of images," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 490-497.

- [14] E. Saber and A.M. Tekalp, "Integration of color, edge and texture features for automatic region-based image annotation and retrieval," *Electronic Imaging*, Vol. 7, Jul 1998, pp. 684–700.
- [15] F. Long, H. Zhang, H. Dagan, and D. Feng, "Fundamentals of content based image retrieval," in D. Feng, W. Siu, H. Zhang (Eds.): "Multimedia Information Retrieval and Management. Technological Fundamentals and Applications," *Multimedia Signal Processing Book*, Chapter 1, Springer-Verlag, Berlin Heidelberg New York, pp.1-26, 2003.
- [16] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, Vol. 38, Issue 11, Nov 1995, pp. 39-41.
- [17] G. Cao, I. Ahmad, H. Zhang, W. Xie, and M. Gabbouj, "Balance learning to rank in Big Data," *22nd European Signal Processing Conference*, 2014.
- [18] G. F. Ahmed and R. Barskar, "A Study on Different Image Retrieval Techniques in Image Processing," *International Journal of Soft Computing and Engineering*, Vol. 1, Issue 4, Sep 2011, pp. 247-251.
- [19] G. Jaswal, A. kaul, "Content Based Image Retrieval – A Literature Review", *National Conference on Computing, Communication and Control*, 2009, pp. 198-201.
- [20] H. H. Wang, D. Mohamad, and N. A. Ismail "Approaches, Challenges and Future Direction of Image Retrieval," *Journal of Computing*, Vol. 2, Issue 6, Jun 2010, pp. 193-199.
- [21] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under Bayesian framework," In *Proc. Third International Conference on Image and Graphics (ICIG)*, Hong Kong, China, 2004, pp. 306-309.
- [22] H. Tamura and N. Yokoya, "Image Database Systems: A Survey," *Pattern Recognition*, Vol. 17, No. 1, 1984, pp.29–49.
- [23] H. Xu, X. Zhou, L. Lin, Y. Xiang, and B. Shi, "Automatic Web Image Annotation via Web-Scale Image Semantic Space Learning," in *Proc. of the Joint International Conferences on Advances in Data and Web Management*, 2009, pp. 211–222.
- [24] J. Eakins and M. Graham, "Content-based Image Retrieval," *University of Northumbria at Newcastle*, 1992.
- [25] J. G. Dy, C. E. Brodley, and S. Wrobel, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, Vol. 5, 2004, pp. 845–889.
- [26] J. Hafner, H. S.Sawhney, W. Equits, M. Flickner, and W. Niblack, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, Jul 1995.
- [27] J. Hou, D. Zhang, Z. Chen, L. Jiang, H. Zhang, and X. Qin, "Web Image Search by Automatic Image Annotation and Translation", in *17th Interna-*

- tional Conference on Systems, Signals and Image Processing, 2010, pp. 105-108.
- [28] J. Raitoharju, H. Zhang, E. C. Ozan, M.A. Waris, M. Faisal, G. Cao, M. Roininen, I. Ahmad, R. Shetty, S. P.C., S. Uhlmann, K. Samiee, S. Kiranyaz, M. Gabbouj, "TUT MUVIS IMAGE RETRIEVAL SYSTEM PROPOSAL FOR MSR-BING CHALLENGE 2014," 2014 IEEE International Conference on Multimedia and Expo, Chengdu, China, Jul. 2014.
- [29] J. R. Smith and S.F. Chang, "Automated binary texture feature sets for image retrieval," In Proc. IEEE Int. Conf_Acoust., Speech, and Signal Proc., (Atlanta), GA, 1996.
- [30] J. R. Smith and S-F Chang, "Tools and techniques for Color Image Retrieval," Symposium on Electronic Imaging: Science and Technology, Storage and Retrieval for Image and Video Databases IV, vol. 2670, San Jose, CA, 1996.
- [31] J. Su, B. Wang, H. Yeh, and V. S. Tseng "Ontology-Based Semantic Web Image Retrieval by Utilizing Textual and Visual Annotations," in Web Intelligence/IAT Workshops, 2009, pp: 425-428.
- [32] J. Weston, A. Makadia and H. Yee., "Label Partitioning For Sublinear Ranking," ICML 2013.
- [33] J. Xu and H. Li, "AdaRank: a boosting algorithm for information retrieval," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007, pp. 391-398.
- [34] K. Klaydios, "Relevance Feedback Methods for Web Image", Dissertation Thesis, Chania, Technical University of Crete, Department of Electronic and computer Engineering, 2004, pages 81.
- [35] K.-M. Wong, L.-M. Po, and K.-W. Cheung, "Dominant Color Structure Descriptor for Image Retrieval," Department of Electronic Engineering, City University of Hong Kong, 2007, pp. 365-368.
- [36] L. Breiman "Bagging predictors," in Machine Learning, Vol. 24, 1996, pp. 123-140.
- [37] L. Page, S. Brin, R. Motwani, and T. Winograd "The PageRank citation ranking: bringing order to the Web," Technical Report, Stanford digital library, Jan 1998.
- [38] M. A. Stricker and M. Orengo "Similarity of color images," in Proc. of SPIE, Storage and Retrieval for Image and Video Database, 1995, pp. 381-392.
- [39] M. Bhuvaneshwari and Mr. P. Narendran, "Review of Recent Trend in Effective Image Retrieval Techniques," Vol. 3, No. 3, ISSN 2249-1945, 2013, pp. 1259-1263.

- [40] M. Grimaldi, P. Cunningham, and A. Kokaram, "An evaluation of alternative feature selection strategies and ensemble techniques for classifying music," in Proc. Workshop on Multimedia Discovery and Mining, 2003.
- [41] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, 1991.
- [42] M. Rastegari, A. Farhadi, and D. Forsyth, "Attribute discovery via predictable discriminative binary codes," in *Vision-ECCV*, 2012.
- [43] M. Saad, "Image Retrieval Literature Survey," *EE 381K: Multidimensional Digital Signal Processing*, Mar 18, 2008.
- [44] M. S. Pal and Dr. S. K. Garg, "Image Retrieval: A Literature Review," *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, Vol. 2, Issue 6, ISSN: 2278 – 1323, Jun 2013, pp. 2077-2080.
- [45] N. Jain & Dr. S. S. Salankar, "Color & Texture Feature Extraction for Content Based Image Retrieval," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)* e-ISSN: 2278-1676, p-ISSN: 2320-3331, 2014, pp. 53-58.
- [46] N. Singhai and Prof. S. K. Shandilya, "A Survey On: Content Based Image Retrieval Systems," *International Journal of Computer Applications (0975 – 8887)*, Vol. 4, No.2, Jul 2010, pp. 22-26.
- [47] O. Guldogan, E. Guldogan, S. Kiranyaz, K. Caglar, and M. Gabbouj, "Dynamic Integration of Explicit Feature Extraction Algorithms Into MUVIS Framework," *FINSIG 2003, Finnish Signal Processing Symposium*, Tampere, Finland 2003.
- [48] P. Jayaprabha and Rm. Somasundaram, "Content Based Image Retrieval Methods Using Graphical Image Retrieval Algorithm (GIRA)," *International Journal of Information and Communication Technology Research*, Vol. 2, No. 1, Jan 2012, pp. 9-14.
- [49] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations, CoRR*, Vol. abs/1312.6229, 2013.
- [50] Prof. S. P. Gaikwad and Dr. L. R. Ragha, "Emotion Detection Using Neural Network for Images," in Proc. *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 9, Sep 2013, pp. 2459-2462.
- [51] Q. V. Le, M. A. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," in Proc. 29th *International Conference on Machine Learning (ICML)*, 2012, pp. 81-88.
- [52] R. Datta, D. Joshi, J. Li, and J. Z. Wang "Image retrieval: ideas, influences, and trends of the new age," *ACM Computing Surveys*, Vol. 40, No. 2, Apr 2008, pp.1 – 60.

- [53] R. He, N. Xiong, L. T. Yang, and J. H. Park, "Using Multi-Modal Semantic Association Rules to fuse keywords and visual features automatically for Web image retrieval," In *Information Fusion*. Vol. 12, Issue 3, 2010, pp. 223-230.
- [54] R. Priyatharshini and S. Chitrakala, "Association based Image retrieval: A survey," Springer-Verlag Berlin Heidelberg, 2013, pp. 17-26.
- [55] R. Smith, "An overview of the tesseract ocr engine," in *ICDAR*, Vol. 7, 2007, pp. 629–633.
- [56] R. S. Kumar and Dr. M. Senthilmurugan, "Content-Based Image Retrieval System in Medical," *International Journal of Engineering Research & Technology (IJERT)*, ISSN: 2278-0181, Vol. 2, Issue 3, Mar 2013.
- [57] S. Brin and L. Page, "The anatomy of a large scale hypertextual Web search engine," *Computer Network and ISDN Systems*, Vol. 30, 1998, pp. 107-117.
- [58] S.-F. Chang, A. Eleftheriadis, and R. McClintock, "Next-generation content representation, creation and searching for new media applications in education," in *Proc. IEEE*, Vol. 86, No. 5, Sep 1998, pp. 884-904.
- [59] S. Gerard and C. Buckely, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management*, Vol. 24, No.5, Jan 1998, pp. 513-523.
- [60] S. Kiranyaz and M. Gabbouj, "Hierarchical Cellular Tree: An Efficient Indexing Scheme for Content-based Retrieval on Multimedia Databases," *IEEE Transactions on Multimedia*, Vol. 9, No. 1, Jan 2007, pp. 102-119.
- [61] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning Multi-scale Block Local Binary Patterns for Face Recognition," In *Proc. of International Conference on Biometrics (ICB)*, 2007, pp. 828-837.
- [62] S. Marinai, "A Survey of Document Image Retrieval in Digital Libraries," University of Florence, Italy.
- [63] S. Siddique, "A Wavelet Based Technique for Analysis and Classification of Texture Images," Carleton University, Ottawa, Canada, Project Report 70.593, Apr 2002.
- [64] T. Mäenpää and M. Pietikäinen, "Texture Analysis With Local Binary Patterns," Department of Electrical and Information Engineering, InfoTech Oulu, University of Oulu, May 2004.
- [65] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol.24, No. 7, Jul 2002, pp. 971–987.
- [66] U. Iqbal, I. D. D. Curcio, and M. Gabbouj, "Semi-supervised person re-identification in videos," in *9th International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, Jan. 2014.

- [67] U. Park and A. K. Jain, "3D Model-Based Face Recognition in Video", in 2nd International Conference on Biometrics, 2007.
- [68] V. Kapur, Dr. P. T. Karule, and Dr. M. M. Raghuvanshi, "Content Based Image Retrieval (CBIR) Using Soft Computing Technique," 2nd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2013) March 17-18, 2013 Dubai (UAE).
- [69] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in Proc. IEEE Int. Conf. on Image Proc., 1997.
- [70] X.-S. Hua, "Looking into MSR-Bing Image Retrieval Challenge," in Microsoft Research Technical Report MSR-TR-2013-76, Apr 2013.
- [71] X.-S. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, and J. Li, "Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines," in Proceedings of the 21st ACM international conference on Multimedia. ACM, 2013, pp. 243–252.
- [72] X.-S. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, and J. Li, "Clickture: A large-scale real-world image dataset," in Microsoft Research Technical Report, MSR-TR-2013-75, Aug 2013.
- [73] X.-S. Hua, M. Ye, and J. Li, "Mining Knowledge from Clicks: MSR-Bing Image Retrieval Challenge," Microsoft Corporation.
- [74] Y. Alemu, J. Koh, M. Ikram, and D. Kim, "Image Retrieval in Multimedia Databases: A Survey," Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, pp. 681-689.
- [75] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," The Journal of machine learning research, Vol. 4, Nov 2003, 933–969.
- [76] Y. -H. Lei, Y. -Y. Chen, B. -C. Chen, L. Lida, and W. H. Hsu, "Where Is Who: Large scale Photo Retrieval by Facial Attributes and Canvas Layout," In Proc. of the 35th International ACM SIGR Conference on Research and Development in Information Retrieval, Portland, Oregon, USA, 2012.
- [77] Y. Rui, A. C. She, and T. S. Huang, "Modified Fourier Descriptors for Shape Representation - A Practical Approach," in Proc. of First International Workshop on Image Databases and Multi Media Search, 1996.
- [78] Y. Rui and T. S. Huang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," Journal of Visual Communication and Image Representation, Vol. 10, 1999, pp. 39–62.
- [79] <http://www.google.com> (Last accessed: 11-01-2014).
- [80] <http://hunspell.sourceforge.net/> (Last accessed: 01-09-2014).
- [81] <http://iac.tut.fi/resources> (Last accessed: 10-01-2014).
- [82] <http://research.microsoft.com/irc2013/> (Last accessed: 12-31-2014).
- [83] <http://www.bing.com> (Last accessed: 11-01-2014).

-
- [84] http://www.code10.info/index.php?option=com_content&view=article&id=49:article_canberra-distance&catid=38:cat_coding_algorithms_data-similarity&Itemid=57 (Last accessed: 12-15-2014).
- [85] <http://www.icme2014.org/msr-bing-image-retrieval-challenge> (Last accessed: 12-10-2014).
- [86] <http://www.mathworks.se/discovery/feature-extraction.html> (Last accessed: 09-01-2014).
- [87] <http://www.pbarrett.net/techpapers/euclid.pdf> (Last accessed: 12-01-2014).
- [88] <http://www.posh24.com/> (Last accessed: 09-05-2014).
- [89] <https://software.intel.com/en-us/vcsourc/tools/perceptual-computing-sdk> (Last accessed: 06-01-2014).
- [90] <http://www.yahoo.com/> (Last accessed: 12-20-2014).