UNIVERSITY of KASDI MERBAH OUARGLA

Faculty of New Technologies of Information and Communication

Department of Computer Science and Information Technologies

Professional Master's Degree

Domain: Computer and Information Technology

Specialty: Computer Science Industrial

Presented by: Aouameur Souraya

Image Retrieval Using Patch Based Feature

Dr. Belal Khaldi	Supervised	UKM Ouargla
Dr. Oussama Aiadi	Examiner	UKM Ouargla
Ms. Farah Dabagh	Examiner	UKM Ouargla

Academic year 2017/2018

Dedicate

I'd like to dedicate my work:

To my beloved parents who were always there for me, giving me support and love and cherish.

To my dearest brothers and sisters for their help to me.

To my coolest friends in real life and those who I've know in social media.

Acknowledgments

My thanks go to all people that were involved in my master's thesis. I would like to thanks my supervisor, Dr Belal Khaldi, for his advice, support, and guidance to conduct my research.
I'd thank the membres of the jury Dr. Aiadi.O and Ms. Dabagh.F for agreeing to judge my work.
I would also like to thank all my friends and my colleagues in the Computer Science and Information Technologies department for their friendship and assistance during my research study.

Table of Contents

Abstract

Image retrieval from large datasets has become an area of wide interest nowadays in many applications. In this thesis we present a content-based image retrieval system that uses patches from image as visual features to describe it. We adopted Dense Micro-Block Difference as local feature in our CBIR system. The used features are based on idea that small patches in a texture image exhibit a characteristic structure and, if captured efficiently, discriminative information can be obtained. Such features are encoded using bag of visual words to obtain an image descriptor which considers higher order statistics. In this step, first, the local features are extracted from the training images and then exemplar features are chosen as the textons (using K-means clustering). These textons are used to label all the features from training and test images. Our experimental evaluation of the system is based on different image datasets. From the experimental results, we found out that DMD significantly outperforms other features.

Keywords: Image Retrieval, DMD, BoVW, Image Feature, Patch.

Résumé

La récupération d'images à partir de grands ensembles de données est devenue de nos jours un domaine d'un grand intérêt dans de nombreuses applications. Dans cette thèse, nous présentons un système de recherche d'images basé sur le contenu qui utilise des correctifs de l'image comme des caractéristiques visuelles pour le décrire. Nous avons adopté la différence dense de micro-bloc comme caractéristique locale dans notre système CBIR. Les caractéristiques proposées sont basées sur l'idée que de petits correctifs dans une image de texture présentent une structure caractéristique et, s'ils sont capturés efficacement, des informations discriminantes. De telles caractéristiques sont codées en utilisant un sac de mots visuels pour obtenir un descripteur d'image qui tient compte des statistiques d'ordre supérieur. Dans cette étape, d'abord, les caractéristiques locales sont extraites des images d'apprentissage et ensuite les caractéristiques exemplaires sont choisies comme textons (en utilisant le clustering K-means). Ces textons sont utilisés pour étiqueter toutes les caractéristiques de l'entraînement et des images de test. Notre évaluation expérimentale du système est basée sur différents ensembles de données d'images. À partir des résultats expérimentaux, nous avons découvert que DMD surpasse significativement les autres fonctionnalités.

Mots clés : Récupération d'image, DMD, BoVW, caractéristiques de l'image, correctif.

ملخص

أصبح استرجاع الصور من مجموعات البيانات الكبيرة مجالا للاهتمام في الوقت الحاضر في العديد من التطبيقات. في هذه الرسالة ، نقدم نظامًا لاسترجاع الصور يعتمد على المحتوى ويستخدم تصحيحات من الصورة كسمات مرئية لوصفها. اعتمدنا فرق الكتل الصغرى الكثيفة كميزة محلية في نظام CBIR لدينا. تستند الميزات المقترحة على فكرة أن بقع صغيرة في صورة نسيج تظهر بنية مميزة ، وإذا تم التقاطها بشكل فعال ، معلومات تمييزية. يتم ترميز هذه الميزات باستخدام كيس من الكلمات المرئية للحصول على واصف الصورة الذي يعتبر إحصائيات ترتيب أعلى. في هذه الخطوة ، أولاً ، يتم استخراج الميزات المحلية من صور التدريب ، ثم يتم اختيار الميزات النموذجية على هيئة النسق (باستخدام K-means clustering). تستخدم هذه النصوص لتصنيف جميع الميزات من صور التدريب والاختبار. يعتمد تقييمنا التجريبي للنظام على مجموعات بيانات صور مختلفة. من النتائج التجريبية ، اكتشفنا أن DMD يتفوق بشكل كبير على الميزات الأخرى.

كلمات مفتاحية : استرجاع الصور, DMD, BoVW, خصائص الصور, Patch.

List of Figures

List of Tables

Chapter I. General Introduction

Chapter I. General Introduction

Image is the most common term for any picture of a person, object, or landscape. It can be painting or pencil drawing, etc.

Digital image is a numeric representation of a two-dimensional image; it could be of a vector or a raster type.

Pixels are the smallest elements in the image, lined up in rows and columns. One pixel holds information about the color of a specific point in the image.

Digital image may be characterized: by pixel per inch, dots per inch, type of image (black & white, color), resolution, image depth, brightness, dynamic range, contrast, saturation, sharpness, image size, and artifact.

I.1 Image Processing

Image processing is any sort of processing where the input is an image, and the output could be an image or a set of properties or parameters associated with the image. In general, image processing techniques include process the image as a two-dimensional signal and applying a signal processing techniques.

Image analysis is when a computer or an electrical device automatically studies an image for obtains useful information. The device is often a computer, but it may also be an electrical circuit, a digital camera, or a mobile phone. The primary stages of Image Analysis:

Preprocessing is a step where requirements are usually clear and simple, such as erasing remnant from images or removing image information that is not in demand for the application.

Involves limiting the data in the spatial domain or converting it to another field called frequency domain via a mathematical transform, then the extraction of features for the analysis process.

The features extracted from the data reduction process are tested and evaluated for use in the application.

I.2 Application of Digital Image Processing

Digital image processing applications are constantly growing through all areas of science and industry [1], including:

- Medicine, such as detecting cancer in an MRI scan.
- Optical character recognition, such as automatic license plate detection.
- Defense.
- Filtering.

I.3 Problem

Images have always been seen as an effective way to deliver visual data in many applications ranging from industry to academia. With technological development, a large amount of images are generated every day. Therefore, managing and indexing images become essential in order to retrieve relevant images effectively and efficiently.

In a small collection of images, simple browsing can identify an image. This is not the case for a large and varied collection of images, where User encounters the image retrieval problem. The conventional method of image retrieval is searching for a keyword that would match the descriptive keyword specified to the image by a human categorizer. Currently under development, even though several systems exist, is the retrieval of images based on their content, called Content Based Image Retrieval, CBIR. While computationally expensive, the results are more accurate than conventional image indexing. Therefore, there exists a tradeoff between accuracy and computational cost. This tradeoff decreases as more efficient algorithms are utilized and increased computational power becomes inexpensive.

The problem includes inserting an image as a query in a software application designed to use CBIR techniques in extracting and matching visual properties. This is done to retrieve images in the database that are visually similar to the query image.

I.4 Proposed Solution

The design and development of effective and efficient CBIR systems are still a research problem, the solution initially proposed was to extract the features of a query image and compare them to those of database images. Features called Dense Micro-block Difference (DMD) were proposed. These features provide a highly descriptive representation of image patches by densely capturing the granularities at multiple levels and orientations. Unlike most previous works on local features, DMD does not involve any quantization, thus retaining full information. DMD has much lower dimensionality and can be calculated using integral image much faster. The used features are encoded using bag of visual words encode method to obtain an image descriptor which considers high order statistics.

I.5 Organization of This Thesis

Continue of the thesis is organized as follows. Chapter 2 we introduce an overview of the Image Retrieval system and its principle plus summarize some of the related works in the topic of CBIR and the primary research issues. In Chapter 3 we talk about techniques used for feature extraction, similarity measure, and indexing structures. The essential technique used for feature extraction in the proposed systems is discussed in Chapter 4. Simulation results and evaluation of the system are detailed in Chapter 5. Finally, concludes our work and suggests future work.

Chapter II.

Image

Retrieval

Chapter II. Image Retrieval

Image retrieval is one of the computer research areas which can be used for browsing, searching and retrieving images from a large image database. Most traditional and common image retrievals utilize some method of adding metadata such as captioning, keywords, or descriptions to the images (i.e., annotation) so that retrieval can be performed over these annotations. In such techniques, to search for images, the user may provide search terms such as keyword.

Other image retrieval techniques analyze the contents of the image rather than the annotation. The term “content” in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself, which are called features. In such techniques, to search for images, the user may provide image file / link and the system will return "similar" images based on content similarity. The similarity used for search criteria can be color distribution in images, texture, and region / shape attributes.

II.1 Text Based Image Retrieval

In text-based image retrieval (TBIR), images are indexed using keywords, topic titles, or classification codes, which in turn are used as keys during search and retrieval. Text-based is subjective because different users use different keywords to annotate images. Sometimes the descriptions of the texts are incomplete because they cannot well describe the complex image content. Examples are texture images that cannot be described by text. Text information about images can be easily searched using current technology, but requires humans to describe each image in the database. This is impractical for very large databases, or for images that are created automatically, for example, from surveillance cameras. It's also possible to miss images that use different synonyms in their descriptions. Systems based on image classification in semantic categories such as "cat" as a subclass of "animal" avoid this problem, but still face the same issues.

II.2 Content Based Image Retrieval

Content-based image retrieval (CBIR) uses image content to search and retrieve digital images. CBIR systems were introduced to address problems associated with text-based image retrieval. CBIR is a set of techniques that retrieve related images from an image database based on automatically derived image features. The main objective of CBIR is the effectiveness during image indexing and retrieval, that way reducing the need for human involvement in the indexing process. The computer must be able to retrieve images from a database without any human assumption on a specific domain.

One of the main tasks of CBIR systems is comparing similarities. Firstly, it extracts features of each image based on its pixel values. These features become the image representations to measure similarity with other images in the database. And then, it compares the submitted image with other images in the dataset by calculating the distance between their corresponding features, the following figure 1 shows the general appearance of the cbir system.

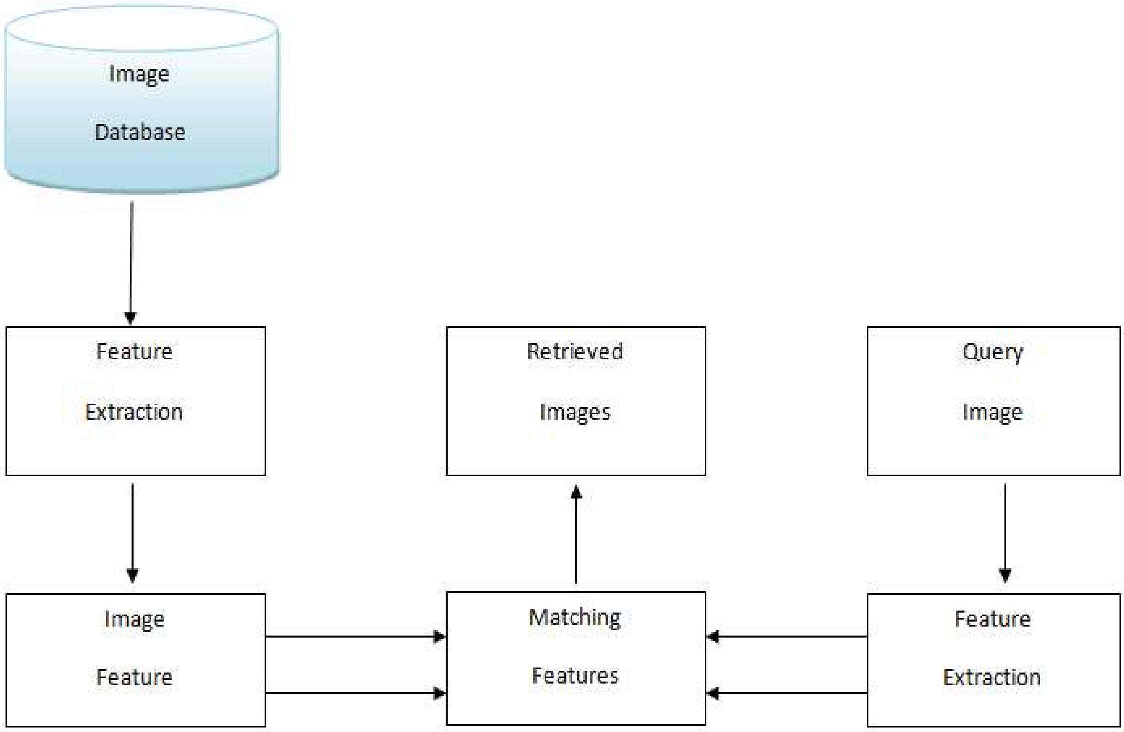

Figure 1. Block Diagram of a General CBIR System.

II.3 Image Distances Measures

In the most common case, the query is a digital image that is compared to images in the database according to the measurement of the feature distance. When the distance returned is zero, this means that the image exactly matches the query. Values higher than zero indicate different degrees of similarity with the query. Image search engines usually return a set of images according to their distance from the query.

II.4 Areas of Application

Image retrieval is extremely useful in a large number of applications such as publishing, advertising, historical research, fashion and graphics design and architecture and engineering design, crime prevention, medical diagnosis, geographical information and remote sensing systems, etc. [2]. A typical image retrieval application example is a design engineer who needs to search his organization database for design projects similar to that required by his clients, or the police seeking to confirm the face of a suspected criminal among faces in the database of renowned criminals. In the commerce department, before trademark is finally approved for use, there is need to find out if such or similar ones ever existed. In hospitals, some ailments require the medical practitioner to search and review similar X-rays or scanned images of a patient before proffering a solution. The most important application, however, is the Web, as big fraction of it is devoted to images, and searching for a specific image is indeed a daunting task. Numerous commercial and experimental CBIR systems are now available, and many web search engines are now equipped with CBIR facilities, as for example Alta Vista, Yahoo and Google [3]. Some of the commonly used CBIR Systems are described as follows:

Photobook developed in the MIT Media Laboratory performs queries based on features associated with images. The features relate to particular models fitted to each image. Commonly the models involve color, texture, and shape, although features from any model may be used (e.g., results of face recognition). Features are compared using one out of the library of matching algorithms such as Euclidean or Mahalanobis distance, Kullback-Leibler divergence, vector space angle (correlation), histogram distance, peaks in Fourier space, wavelet tree distance, or user-defined algorithms, as well as any linear combination of these [4]. The face recognition technology of Photobook has been used by Viisage Technology in a FaceID package, which is used in several US police departments.

In **Alexandria Digital Library (ADL)** CBIR system [5], the only texture feature is extracted from images using texture features that can be explored by images. With the help of browser map the user can interrelate magnifies a two dimensional world map to find its significance area, and choose a question area that has to involve with the database images. Right now, a list of images, query parameters can be selected by the user for example, aerial of the photo, map, images of remote sensor etc. and then with the help of browser map the images that are overlapped with area are retrieved. An important focus for ADL's collection is on information supporting basic science, including the Earth and Social Sciences. The image datasets (will) include digital elevation models (DEMs), digital raster graphics (DRGs), scanned aerial photographs, Landsat images, seismic datasets, Sierra Nevada Ecologic Project datasets, and Mojave Ecologic Project datasets.

VIR Image Engine system [6] is an expandable framework structure for building CBIR systems. In this CBIR system, the main concept is the feature type, computation and distant matching**.** There are five abstract data types: a) global value, b) color histograms, c) local value, d) histograms and e) graphs. The VIRAGE system gives a set of general primitives. The VIRAGE system gives a GUI tool set essential for the growth of a GUI edge. These contain services for placing of images, image queries and maintain many popular formats of image file. Query by sketch is also used where picture can be sketch by user with drawing tools and color palette. The system was integrated into databases from Sybase, Object Design, and Objectivity, and added as a component to the Oracle DBMS. The AltaVista Photofinder and Illustra's Visual Intelligence system are two applications of Virage technology.

Chapter III.

Image Features

Chapter III. Image Features

To best of our knowledge, there is no comprehensive or accurate definition of what constitutes an image feature. It often depends on the problem it tries to resolve or on the application area. The feature is defined as an "interesting" part of the image, and is used as a starting point in the main priorities of the algorithms. Often the general algorithm is only as good as the detector features. Therefore, the desired property of the feature detector is its repeatability: whether the same feature is detected in two or more different images for the same scene. The most important feature types that can be observed when trying to recognize images are spatial, temporal and textural. The recognition/retrieval systems are usually divided into three main phases: extraction, selection and classification. For an appropriate representation, there must be a rational relation between the features that are most important. The quality of image feature directly affects the performance of the classification system. The final result of feature extraction process is a set of values, commonly called a feature vector, which constitutes the representation of an image. Image features could be broadly categorized into:

• **General features:** Independent features such as color, texture and shape. According to the level of abstraction, general features can also be divided into:

- *Pixel level features:* Features are calculated in each pixel, for example, color and location.

- *Local Features:* After splitting the image into a set of sub-regions, Local features are calculated from each of these sub-regions.

- *Global Features:* Features calculated on the full image or just an ordinary sub-region of the image.

• **Range-specific features:** Applied on attached features such as human faces, fingerprints, and conceptualization.

All features can be roughly graded to low-level features and high-level features. Low-level features can be extracted directly from the image, while extracting a high-level feature relies on low-level features.

III.1 Color Features

Color is one of the most commonly used features in image retrieval. Color is the first and most direct visual feature for indexing and retrieving images. The most common color descriptors are Color Moments, Color Histogram, Color Coherence Vector and Color Correlogram. Color plays a very important role in the mechanism of human visual perception. All methods of representing the color feature of the image can be classified into two groups: color histograms and statistical methods for color representation. The most commonly used color spaces are: RGB (red, green, and blue), CMY (cyan, magenta and yellow), CMYK (Cyan, Magenta, Yellow, and Black), Lab (Lightness, and b are two color dimensions) and HSI, HSV (hue, saturation, and value). Color features have many advantages such as durability, effectiveness, simplicity of implementation, computational simplicity and low storage requirements.

The following table lists the advantages of using different color features.

Color Features	Advantages
Color Moments	Compact features of color and sensitive to spatial information.
Color Histogram	Extracts both local and global features of colors.
Color Coherence Vector	More efficient results.

Table 1. Color Features and its advantage of each one. **[7]**

III.2 Texture Features

Texture is a significant image property and is a strong regional descriptor that helps in retrieval process. The texture, on its own, has no ability to find similar images, but it can be used to classify textured images from non-textured images and then combine them with another visual feature such as color to make the retrieval more efficient.

Texture gives us information about the structural arrangement of the surfaces and objects on the image. This depends on the density distribution of colors within the image, rather than separated pixels. The most common statistical features include co-occurrence matrix-based features, general statistical features calculated from the pixel density values, and texture histograms based on Tamura features. Grey Level Co-occurrence Matrices (GLCM) proposed by Haralick [8] is an effective way to represent image texture features. Texture analysis by Gabor filters [9] is a special case of the wavelet method. This is the most commonly texture feature used by CBIR systems. In most of CBIR systems based in Gabor Wavelet, the mean and standard deviation of wavelet distributions conversion uses coefficients to construct the feature vector. A new feature scheme called Enhanced Gabor Wavelet Correlogram (EGWC) [10] has been proposed for enhanced image indexing and retrieval.

The following table illustrates the advantages of different texture features.

Texture features	Advantages
Tamura	Provides effective result.
Gabor Filter	Used to detect different Frequency and orientation
Wavelet Transform	Filters with salient Point features, efficient for retrieval.

Table 2. Texture Features and its advantage of each one**. [7]**

III.3 Shape Features

Shape is an important visual feature and one of the basic features for describing image content. The description of the shape cannot be accurately defined. In addition, measuring similarities between shapes is extremely difficult. Therefore, there are two steps necessary to retrieve images based on the shape, namely: the extraction feature and measure the similarity between the extracted features.

Shape features are often used to compare images, along with color and texture features. To represent the shape of an object, different methods are used. Shape representations are divided into two groups: the external (representing the boundary) and the internal (representing the pixels that comprise the region). In addition, they are classified as (a) structural and (b) global. The global boundary descriptors include many features, Fourier descriptors, [11] and wavelet descriptors. Regions can be described in terms of the simplest geometrical features, such as an area or compactness. The Grid based method is the most common way to describe the shape of an object. While, moment invariants are currently the most popular and widely used features for describing regions.

III.4 Similarity Measure

For each image, color, texture and shape features are extracted, described by vectors, and stored in the database. Given a query *q*, the same set of features are extracted, and then matched (i.e., calculate distance) with the already stored vectors in the feature database. Dimensional reduction techniques are sometimes used to reduce calculations. The distances are then sorted in increasing order. Finally, the *N* first images from the sorted list are shown as relevant. Regarding the distance measurement, a one-to-one matching scheme can be used to compare the query and the target image. As instances, we mention the following matching schemes:

The Euclidean distance is the distance between two points in Euclidean space. The two points *P* and *Q* in two dimensional Euclidean spaces and *P* with the coordinates *(p1, p2)*, *Q* with the coordinates *(q1, q2)*. The line segment with the endpoints of *P* and *Q* will form the hypotenuse of a right angled triangle. The distance between two points *p* and *q* is defined as the square root of the sum of the squares of the differences between the corresponding coordinates of the points. The two-dimensional Euclidean geometry, the Euclidean distance between two points *a = (ax, ay)* and *b = (bx, by)* is defined as

$$d(a,b) = \sqrt{(bx-ax)^2+(by-ay)^2}$$

The chamfer distance [12] relatively well approximates the Euclidean distance and is widely used because of its relatively small computational requirements as it imposes only 2 scans of the n-dimensional image independently of the dimension of the image. The chamfer distances are widely used in image analysis of the Euclidean distance with integers. The chamfer distance *dM* between 2 points *A* and *B* is the minimum of the associated costs to all the paths *PAB* from *A* to *B*

$$d_M(A,B) = \min_{P_{AB}} W(P_{AB})$$

The distance between two points in a grid is based on a strictly horizontal and/or vertical path as opposed to the diagonal.

The Manhattan distance [13] is the simple sum of the horizontal and vertical components, whereas the diagonal distance might be computed by applying the Pythagorean Theorem. It is also called the L_1 distance and of $u = (x_1, y_1)$ and $v = (x_2, y_2)$ are two points, then the Manhattan distance between *u* and *v* is given by

$$MH(u,v) = |x_1 - x_2| + |y_1 - y_2|$$

The Earth Mover Distance [14] is the discrete way of writing the famous problem of optimal transport, also called the Wasserstein metric or Monge-Kantorovich. It is a distance between probability density functions, or, on discrete data, histograms.

Two histograms *P* and *Q* are given, as well as a distance affinity matrix *D(i, j)*. This matrix computes the cost of transporting one element of mass (i.e. one pixel) of the *i−th* bin of *P* to the *j −th* bin of *Q*. It computes a flow matrix *F* where *F(i, j)* is the amount of mass in the *i−th* bin of histogram *P* transported to the *j−th* bin of histogram *Q*. The goal of optimal transport is then to find *F* that minimizes the cost of every transports *D(i, j)* to warp histogram *P* to histogram *Q*.

$$EMD(P,Q) = \min_{F} \sum_{i,j} F(i,j)D(i,j)$$

Chapter IV.

Patch-Based Feature

Chapter IV. Patch-Based Feature

Image processing using local patches has become very popular and was seen to be extremely effective and useful. A patch is a small area of pixels which is called as a window. Patches are powerful primitives in the area of Image Processing. The image models enable us to systematically develop algorithms for accomplishing a particular image-related application [16].

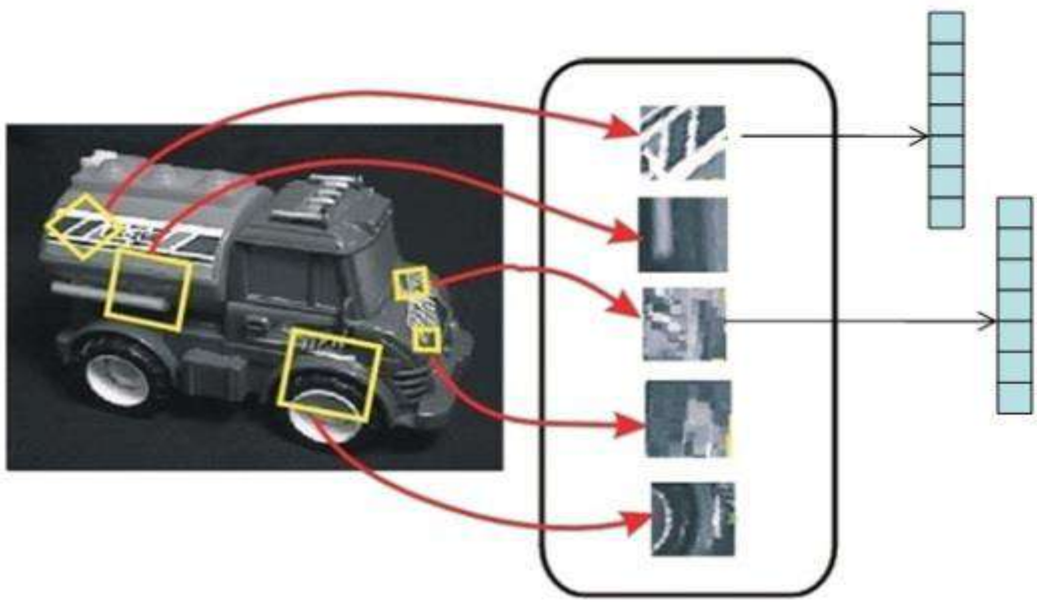

Figure 2. Extracting descriptors from detected keypoints. [26]

There are basically two types of patch based image models: descriptive and generative. *Descriptive* models focus on the extraction of the distinctive features from the given image so that they can facilitate the task of classifying the image into one of several classes. So we can say that they are suitable for the task of classification and recognition. *Generative* models preserve the information in an image that is why they are more desirable for the task of compression and restoration [17] [18].

The main concept of the patch-based processing is, extract all the patches which are very small compared to the original image size, with overlaps from the given image. Patches can either be extracted densely or at interest points. Then the interrelations between those patches are found out. To find the relations between patches, different ways are used: a weighted average of pixels with similar surrounding patches, clustering the patches into disjoint sets and processing each set different, seeking a representative dictionary for the patches and using it to sparsely represent them, gathering groups of similar patches and applying a scarifying transform on them. The manipulated patches are then put back to form the resulting image [19] [20].

In the patch- based method, there is an expectation that every patch taken from the image may find similar ones elsewhere in the image and a joint treatment of these patches may support the reconstruction process which enables better recovery of the image.

IV.1 Patch descriptors

The most popular patch descriptors are:

Histograms of Oriented Gradients (HOG) based descriptors family. Notable examples of this family are SIFT and SURF. A Scale-Invariant Feature Transform (SIFT) [27] descriptor is a 3-D spatial histogram of the image gradients in characterizing the appearance of a keypoint. Speeded Up Robust Features (SURF) [28] descriptor is a 2D Haar wavelet obtained by an integer approximation to the determinant of Hessian matrix that extracts blob-like structures at locations where the determinant is maximum. SIFT and SURF are based on histograms of gradients. That is, the gradients of each pixel in the patch need to be computed. These computations cost time. Even though SURF speeds up the computation using integral images, this still isn’t fast enough for some applications. This is where Binary descriptors come in handy. In general, Binary descriptors are composed of three parts: A sampling pattern, orientation compensation and sampling pairs. Consider a small patch centered around a keypoint. We’d like to describe it as a binary string.

Binary Robust Independent Elementary Features (BRIEF) [29] is the simplest of the methods. It uses a sampling pattern consisting of 128, 256, or 512 comparisons (equating to 128, 256, or 512 bits), with sample points selected randomly from an isotropic Gaussian distribution centered at the feature location.

Oriented FAST and Rotated BRIEF (ORB) [30] overcomes the lack of rotation invariance of BRIEF. ORB calculate a local orientation through the use of an intensity centroid [31], which is a weighted averaging of pixel intensities in the local patch assumed not be coincident with the center of the feature. The orientation is the vector between the feature location and the centroid. While this may seem unstable, it is competitive with the single orientation assignment employed in SIFT [32].

Binary Robust Invariant Scalable Keypoints (BRISK) [33] provides both scale and rotation invariance. In order to compute the feature locations, it uses the AGAST corner detector [34], which improves FAST by increasing speed while maintaining the same detection performance. For scale invariance, BRISK detects keypoints in a scale-space pyramid, performing non-maxima suppression and interpolation across all scales.

IV.2 Patch similarity measures

The development of image patches has been the focus of much work. However, the question of appropriate similarity measure between the patches has largely remained unattended. Typically, a standard distance measure is used for any representation that is used.

IV.2.1 Pixel-based distance

Normalized correlation between patches x1 and x2 is determined as

$$NC(x_1, x_2) = \frac{\sum_d \left(x_{1(d)} - \overline{x_1}\right)\left(x_{2(d)} - \overline{x_2}\right)}{\sigma_1 \sigma_2}$$

, Where $\overline{X}i$ and σi are the mean and standard deviation of pixels in Xi. Because of the factoring in of the means it is much more robust than the pixel-wise distance. Normalized correlation has been used widely in fragment-based recognition [21][22], where it is presumed that viewing conditions are steady, or Instead of that there exist examples from all viewing conditions–in other words, not matching a patch to a version of itself rotated by 90 degrees is acceptable. We would like to avoid such an assumption.

IV.2.2 Probabilistic matching

A different process is taken through some methods that instead of measuring the distance between the representation patches, directly evaluate the probability that the two patches belong to the same class.

This is usually limited to models where a fixed number of patch classes, called parts, are integrated into some framework. A well-known example of this is the family of planet models constellation models [23] [24].

IV.2.3 Descriptor distance

Another common method is to calculate the descriptor of each patch, and then simply apply a distance measure to the descriptors. Often, the descriptors are vectors in a metric space of fixed dimensions and the selection distance is L1. Matching with SIFT descriptors may be the most popular example of such an approach [25].

The common stages for using patch descriptors are:

1. Detect keypoints in image (distinctive points such as angles).
2. Describe each region around a keypoint as a vector of features, using a descriptor.
3. Use descriptors in the application (to compare the function of the descriptors - the use of distance or similarity).

IV.3 Dense Micro-block Difference (DMD)

The used features are based on idea that small patches in a texture image exhibit a characteristic structure and, if captured efficiently, discriminative information can be obtained. Based on the ample evidence from the related work, we use the intensity difference from an image patch to capture the variations in it. Unlike the earlier works, which consider the pixel difference from a circular geometry, it randomly selects the pixel coordinates. Furthermore, individual pixels are more susceptible to noise and do not capture regional information; therefore it uses small blocks in image patch instead of raw pixel values.

IV.3.1 Feature Extraction

To encode the local structure of the patch, it takes the pairwise intensity differences of smaller blocks in the image patch. It addresses smaller square blocks in an image patch as “micro-blocks” and their average intensity is considered to capture variations. The micro-blocks are smaller square regions inside the patch. A large number of micro block pairs aids in obtaining a rich and discriminating representation of the patch by capturing the variations in different directions and scales. The vectors obtained from different size micro blocks are stacked together to capture information at multiple resolutions. It can be observed that the intensity difference is taken in different directions, unlike, e.g. LBP [35] and SRP [36], which only consider the radial direction. Furthermore, the distance between the micro-blocks is not constant, thus, the variations in a patch are captured at different scales.

Formally, given a patch p of size $L \times L$ and two sets of sampling points

$$X = \{x_1, x_2, .., x_N\},\ Y = \{y_1, y_2, .., y_N\}$$

, the DMD for the micro-blocks of size s is defined by:

$$v_s(p) = \{M_s(x_1) - M_s(y_1), \dots, M_s(x_N) - M_s(y_N)\}$$

, where $M_s(x)$ is the average intensity of the pixel in the micro block located at the position $x = (a, b)^T$ in the patch and is given as

$$M_s(a, b) = \sum_{i=0}^{s-1} \sum_{j=0}^{s-1} p(a + i, b + j)\ /s^2$$

, where *p (a, b)* denotes the pixel intensity in the patch p at the location *{a, b}*. The features are normalized with the *l2* norm of the patch to reduce the effect of the illumination changes. The formulation is given for a single micro-block resolution s, however for multiple resolutions it can be easily extended by concatenating the features of different micro-blocks sizes.

The feature is completely specified by the following parameters: *X*, *Y*, *L* and s. The sampling points sets *X* and *Y* determine the location of micro-blocks in the patch and play an important role in the design of the descriptor. There are a lot of different spatial arrangements for selecting the sampling point for keypoint matching [37]. In this case the points are selected from isotropic Gaussian distribution. Once selected the micro-block coordinates *(X, Y)* are kept fixed, so that the same pixels are considered for the feature computation from every patch. In this arrangement the coordinates are more densely distributed towards the center of the patch than towards its boundaries. Thus, a larger weight is given to the center than to its boundaries, like for SIFT features. The randomness in the sampling points helps in capturing the variations at different scales because the distance between the sampling points is not constant. Moreover, the magnitude of the difference without any thresholding helps in retaining the discriminative power of the features.

The following figure illustrates the image patch and the micro-blocks. The outer grid represents image patch of 21 × 21 and the gray and white square are micro-blocks. Three different micro-block resolutions are shown. Multi-resolution features are concatenated together to get the final feature vector.

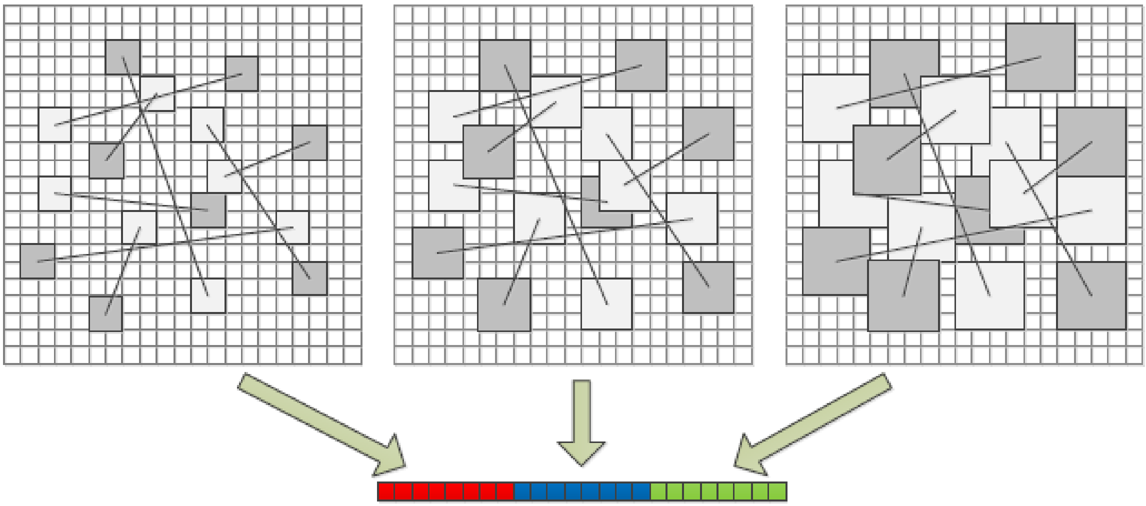

Figure 3. The micro-blocks pairs in an image patch. [38]

IV.3.2 Bag of Visual Words

In the computer vision, the bag of visual words (BoVW Model) [39] can be applied to image classification, by manipulating image features as words. In the document classification, the bog of words is a sparse vector for a number of repeating words. That is, a spare histogram on the vocabulary. In computer vision, a bag of visual words is a vector of the number of occurrences of vocabulary of local image feature.

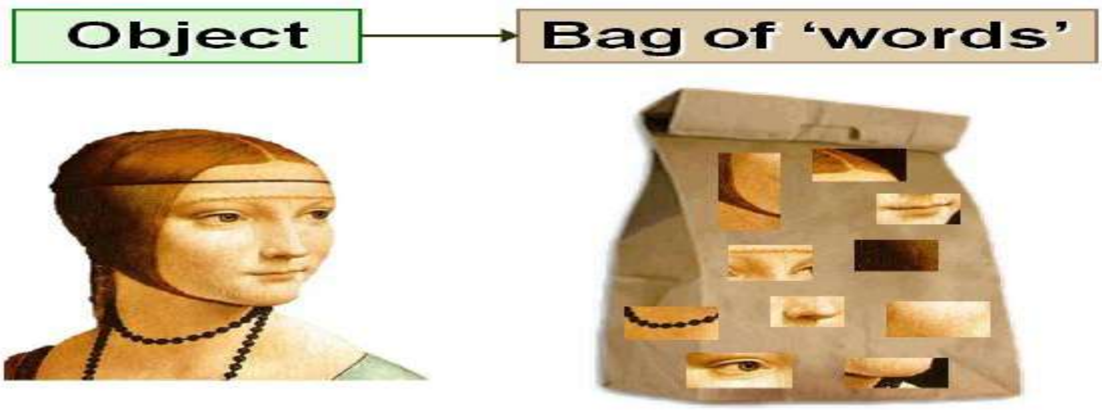

Figure 4. Illustration of Bag of words model in images. [40]

To represent an image using the BoW model, an image can be treated as a document. Similarly, "words" should be defined in images as well. To do this, it usually includes the following three steps: feature detection, feature description, and codebook generation. [41] A definition of the BoW model can be the "histogram representation based on independent features". [42] Content based image indexing and retrieval (CBIR) appears to be the early adopter of this image representation technique. [43]

After feature detection, each image is extracted through many local patches. Feature representation methods deal with how to represent patches as numerical vector. These vectors are called feature descriptors. A good descriptor must have the ability to handle intensity, rotation, scale and affine variations to some extent.

IV.3.2.1 Codebook Generation

The final step for the BoW model is to convert vector-represented patches to "codewords" (analogous to words in text documents), which also produces a "codebook" (analogy to a word dictionary). A codeword can be considered as the representative of many similar patches. One simple way is to perform k-means clustering on all the vectors. Codewords are then identified as centers of the learned clusters. The number of the clusters is the size of the codebook (similar to the size of the word dictionary).

Thus, each patch in an image is set to a specific codeword through the clustering process. An image can therefore be represented as a vector of visual words showing how many times each of the visual words occurs in the image. Figure 5 shows in a schematic fashion how the visual vocabulary is constructed and the vector of visual words (BoVW) is assigned.

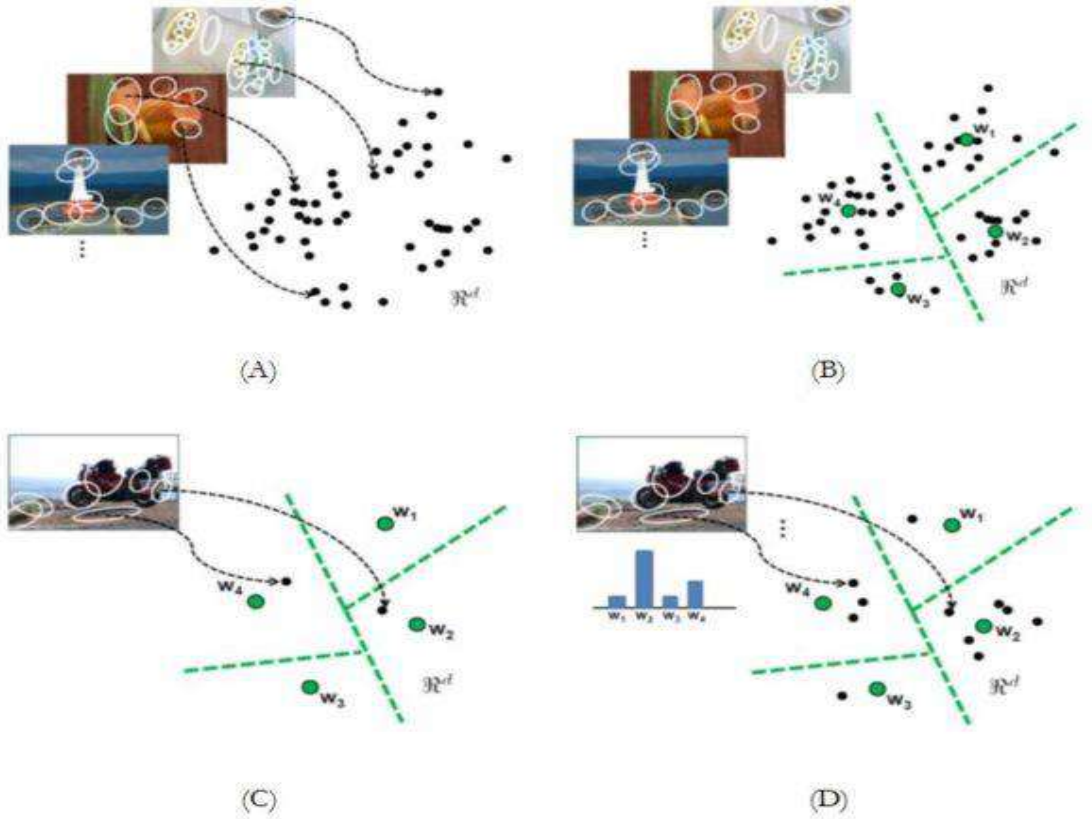

Figure 5. Illustration of visual vocabulary construction and word assignment. [46]

IV.3.3 Distance Measure

In the BoVW approach, the similarity between a vector ***A*** and a query vector ***B*** can be determined by computing the cosine of the angle between them.

Cosine similarity is the measurement of similarity between non-zero vectors of an internal product space that measures cosine of the angle between them. A 0 ° C is 1, which is less than 1 for any other angle in the interval [0, 0.5π].

Cosine similarity is the measurement of similarity between non-zero vectors of an internal product space that measures cosine of the angle between them.

$$cosine(A,B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Hence a judgment of orientation and not magnitude.: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90 degrees have a similarity of 0, and two vectors diametrically opposed have a similarity -1, independent of their magnitude. Cosine is particularly used in positive space, where the result is precisely determined in [0,1]. The name is derived from the term "cosine direction": in this case, note that the unit vectors are "very similar" if they are parallel and "very dissimilar" if they are perpendicular (orthogonal). This is similar to the cosine, which is unity (maximum value) when the segments subtend a zero angle and zero (unattended) when the segments are perpendicular.

Chapter V. Evaluation and Discussion

Chapter V. Evaluation and Discussion

In this chapter, we will develop our application; we will also present the implementation of our application using the language Matlab. First, start with a presentation of the chosen programming language. Then we present screenshots of the execution of our application, explaining the experimental setup. Last, we will report the obtained results and discuss them.

V.1 Presentation of Development Tools

The material used is a personal PC Acer I3 with a 4GB memory capacity, and an Intel® core™ i3 1.70 GHz, with Windows 7 Professional, service pack 1 64 bit type system.

MATLAB (Lab Matrix) is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks, MATLAB allows manipulation of arrays, collocation of functions and data, implementation of algorithms, creation of user interfaces, and interaction with programs written in other languages, including C, C ++, C #, Java, Fortran and Python.[44]

App Designer is a rich development environment that provides layout and code views, a fully integrated version of the MATLAB editor, and a host of interactive components. You can package an app installer file directly from the App Designer tool strip, or you can create a standalone desktop or web app (requires MATLAB Compiler™).[45]

V.2 Experimental Setup

V.2.1 Databases

In our thesis we used two databases:

- **The Columbia Object Image Library (COIL-100)** is a database of 100 color images objects. The objects were placed on a turntable with a motor on a black background. The turntable was rotated through 360 degrees to vary object pose with respect to a fixed color camera. Images of the objects were taken at pose intervals of 5 degrees. This corresponds to 72 poses per object. The images were size normalized. COIL-100 is available online. It has constructed a database of 7,200 color images of 100 objects (72 images per object).

The objects have a wide variety of complex geometric and reflectance characteristics as seen in **Figure 6.**

Figure 6. Representative sample from Coil-100 images.

- The **Corel-10k** dataset contains 100 categories, and there are 10,000 images of different content. Each category contains 100 images in the JPEG formatCorel-5K dataset consists of the first 5000 images, and Corel-10K dataset consists of the 10,000 images. **Figure 7** shows a representative sample from the collected images.

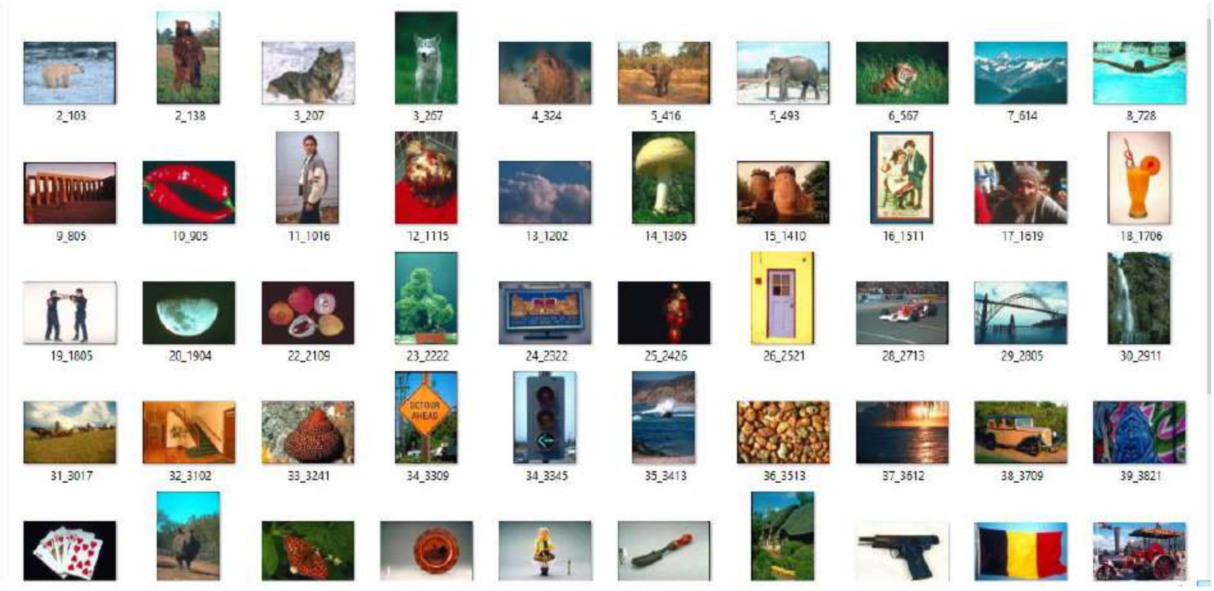

Figure 7. Representative sample from Corel-10K images.

V.2.2 Evaluation Metrics

Many different methods for measuring the performance of a system have been created and used by researchers. The most common evaluation methods namely, Precision and Recall are used to evaluate the retrieval results. Precision and recall is calculated as:

$$Precision = \frac{Number\ of\ Relevant\ Images\ Retrieved}{Total\ Number\ of\ Images\ Ritrieved} \times 100$$

$$Recall = \frac{Number\ of\ Relevant\ Images\ Retrieved}{Number\ of\ Relevant\ Images\ in\ The\ Database} \times 100$$

V.2.3 Involved features:

We added two other image features to test our system:

- **Color Histogram:** A color histogram is a type of bar graph, where the height of each bar represents an amount of particular color of the color space being used in the image. The bars in a color histogram are named as bins and they represent the x-axis. The number of bins depends on the number of colors there are in an image. The number of pixels in each bin denotes y-axis, which shows how many pixels in an image are of a particular color. The color histogram can not only easily characterize the global and regional distribution of colors in an image, but also be invariant to rotation about the view axis.
- **Color Moment:** color moments basis lies in the presumption that the distribution of color in an image can be explained as a probability distribution. Probability distributions are defined by a number of specify moments (for example, normal distributions are known by their mean and variance). Thus, if the color in the image follows a certain probability distribution, the moments of that distribution can be used as features to determine that image based on the color.

V.3 Experimental Results

The main window of our application is:

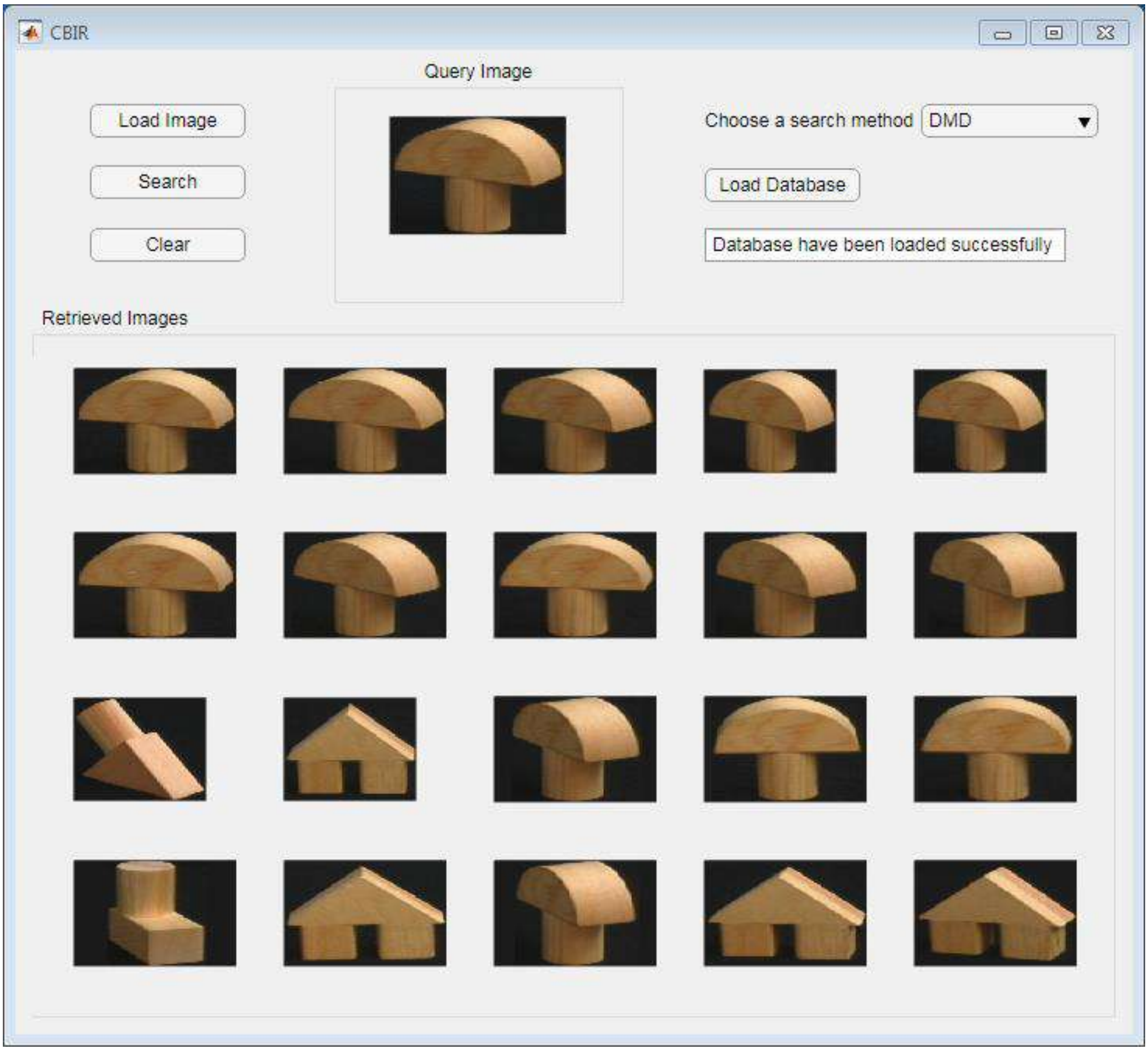

Figure 8. Application interface.

To further evaluate our proposed CBIR system, we split the two datasets into indexing (training) part and test part, for each image in the test part perform a retrieval (Queries) for each query the precision of the retrieval at scope is obtained by gradually increasing the number of retrieved images. The retrieval results are averaged to give the final precision/recall chart of each dataset and it’s showed in Figures 9 and 10.

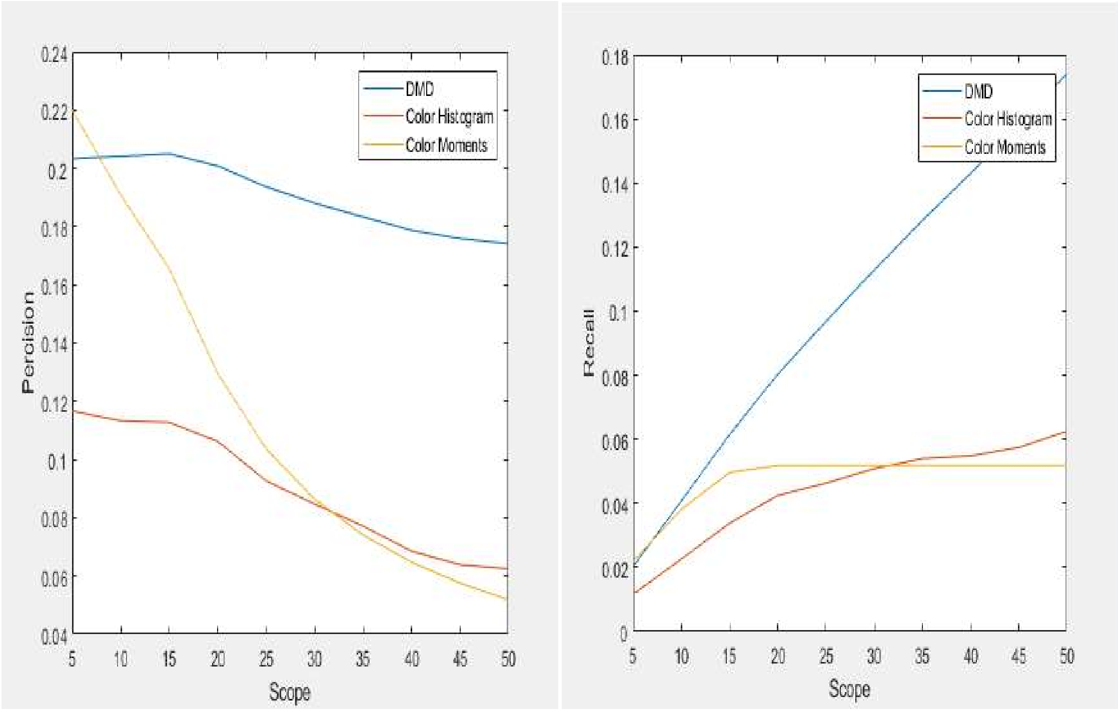

Figure 9. The average Precision\Recall chart of DMD, Color Histogram and Color Moments features in coil dataset.

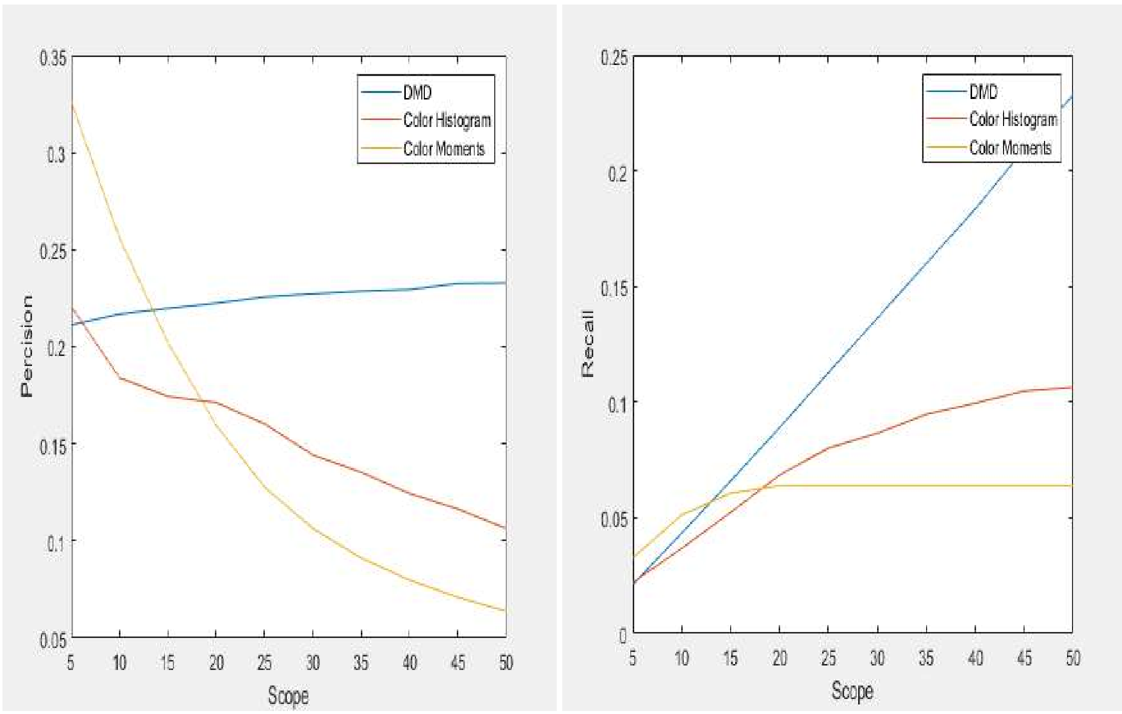

Figure 10. The average Precision\Recall chart of DMD, Color Histogram and Color Moments different features in Corel dataset.

From Figure 9 and Figure 10 it can be noted that the DMD has good average precision and outperforms the Color Histogram and Color Moments features in both datasets.

Conclusion

Content based image retrieval is a challenging method of capturing relevant images from a large storage space. Although this area has been explored for decades, no technique has achieved the accuracy of human visual perception in distinguishing images. Whatever the size and content of the image database is, a human being can easily recognize images of same category.

From the very beginning of CBIR research, similarity computation between images used either patch based or global based features. Using local patches has become very popular and was seen to be extremely effective and useful. We use Dense Micro-block Difference, which is a powerful extraction technique either in describing the content of image patch. Bag of Visual Words is combined with DMD to obtain the final descriptor.

The result obtained during the test phase confirms the effectiveness of our approach.

Our work is only in its initial version; we can say that this work remains open for work of comparison and / or hybridization with other retrieval methods.

References

1. Karol Kozak “management of large sets of data: Capture, Databases, Image Processing, Storage, Visualization” 2014.
2. V. Gudivada and V. Raghavan, “Content-based image retrieval systems,” IEEE Computer, vol. 28, no 9, pp18-22, Sep. 1995.
3. M. L. Kherfi, D. Ziou, and A. Bernardi, “ Image Retrieval From the World Wide Web: Issues, Techniques, and Systems,” ACM Computing Surveys, vol. 36, no. 1, pp. 35–67, March 2004.
4. A. Pentland, R. W. Picard, S. Sclaroff, ”Photobook: Tools for Content-Based Manipulation of Image Databases” Proceedings of SPIE, Storage and Retrieval for Image and Video Databases II, Vol. 2185, pp. 34--47, April (1994).
5. B. S. Manjunath 1995. Image Browsing in the Alexandria Digital Library (ADL) Project D-Lib Magazine, vol. 1(2).
6. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain and C. F. Shu, 1996. Virage image search engine: an open framework for image management. In Electronic Imaging: Science & Technology (pp. 76-87). International Society for Optics and Photonics.
7. Alphonsa Thomas, K. Sreekumar “A Survey on Image Feature Descriptors-Color, Shape and Texture,” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014, 7847-7850.
8. Robert M. Haralick. ”Statistical and structural approaches to texture,” Proc. *IEEE*, vol. 67, no. 5, pp. 786-804, 1979.
9. Gabor, D. (1946). Theory of communication. Journal of the Institute of Electrical Engineers, 93, 429–457.
10. H. Abrishami Moghaddam, M. Nikzad Dehaji. “Enhanced Gabor wavelet correlogram feature for image indexing and retrieval,” Journal Pattern Analysis & Applications Volume 16 Issue 2, May 2013.
11. C.T. Zahn, R.Z. Roskies. "Fourier descriptors for plane close curves," IEEE Trans. Computers, Vol C-21, March 1972, pp. 269-281.
12. Butt MA, Maragos P. 1998. Optimum design of chamfer distance transforms. IEEE Trans Image Process 7(10):1477–1484.
13. Minkowski, Hermann (1910), *Geometrie der Zahlen*, Leipzig and Berlin: R. G. Teubner.

14. F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *J. Math. Phys.*, 20:224-230, 1941.
15. R. Balu, T. Devi "Design and development of automatic appendicitis detection system using sonographic image mining 2012".
16. Idan Ram, Michael Elad And Israel Cohen "Image Denoising Using Nl-Means Via Smooth Patch Ordering," Acoustics, Speech And Signal Processing (Icassp), 2013 IEEE International Conference On, May 2013,Pp. 1350-1354.
17. J. Mairal, M. Elad, G. Sapiro et al., "Sparse representation for color image restoration," IEEE Transactions on Image Processing, vol. 17, no. 1, p. 53, 2008.
18. J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 2272–2279.
19. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by sparse 3-d transform-domain collaborative filtering," IEEE Trans. Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.
20. T. H. Cormen, "Introduction to algorithms". The MIT press, 2001.
21. S. Agarwal and D. Roth. Learning a Sparse Representation for Object Detection. In European Conference on Computer Vision, 2004.
22. Evgeniy Bart and Shimon Ullman. Class-based matching of object parts. In Proceedings of CVPR Workshop on Image and Video Registration, 2004.
23. Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In European Conference on Computer Vision, Freiburg, Germany, 1998.
24. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 264–271, June 2003.
25. D. G. Lowe. Object recognition from local scale-invariant features. Pages 1150-1157, Corfu, Greece, December 2000.
26. Levi Gil, (2013, August 18). A Short introduction to descriptors [blog post], https://gilscvblog.wordpress.com/2013/08/18/a-short-introduction-to-descriptors/
27. Lowe, David G. "Object recognition from local scale-invariant features."*Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. IEEE, 1999.

28. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features.” *Computer Vision–ECCV 2006*. Springer Berlin Heidelberg, 2006. 404-417.
29. Calonder, M., Lepetit, V., Fua, P.”BRIEF: Binary Robust Independent Elementary Features”. ECCV, 778-792 (2010).
30. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.” ORB: An Efficient Alternative to SIFT or SURF”. ICCV, 2564-2571 (2011).
31. Rosin, P. L.: Measuring Corner Properties. Comp. Vis. and Image Understanding, 291-307 (1999).
32. Gauglitz, S., et al.: Improving Keypoint Orientation Assignment. BMVC, (2011).
33. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust Invariant Scalable Keypoints. ICCV, 2548-2555 (2011).
34. Mair, E., et al.: Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. ECCV, 183{196 (2010).
35. T. Ojala, M. Pietikainen, and T. Maenpaa, “Multi resolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, 2002.
36. L. Liu, P. Fieguth, G. Kuang, and H. Zha, “Sorted random projections for robust texture classification,” in IEEE International Conference on Computer Vision (ICCV), 2011, pp. 391–398.
37. M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in Computer Vision–ECCV 2010. Springer, 2010, pp. 778–792.
38. R. Mehta and K. Egiazarian, “Texture Classification using Dense Micro- Block Difference,” IEEE Transactions On Image Processing, Vol. 25, No. 4, April, 2016.
39. Csurka, Gabriella, et al. “Visual categorization with bags of keypoints.”Workshop on statistical learning in computer vision, ECCV. Vol. 1. 2004.
40. Li Fei-Fei, Rob Fergus, Antonio Torralba. “Recognizing and Learning Object Categories” ICCV 2005.
41. Fei-Fei Li; Perona, P. (2005). "A Bayesian Hierarchical Model for Learning Natural Scene Categories". *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2: 524.
42. L. Fei-Fei; R. Fergus & A. Torralba. "Recognizing and Learning Object Categories, CVPR 2007 short course".

43. Qiu, G. (2002). "Indexing chromatic and achromatic patterns for content-based color image retrieval" . *Pattern Recognition*. 35 (8): 1675–1686.
44. Wikipedia, “MATLAB”. https://en.wikipedia.org/wiki/MATLAB
45. MathWorks, “App Designer”.https://www.mathworks.com/help/matlab/app-designer.html
46. Dr. Diaz, D. Santika, (10 May 2017). “IMAGE SEARCH BY CONTENT USING BAG OF VISUAL WORDS PARADIGM” [blog post], http://socs.binus.ac.id/2017/05/10/image-search-by-content-using-bag-of-visual-words-paradigm/