CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

Feature based extraction describes the process of retrieving characteristic features from the images to identify the different traits of the image according to their various low level features like color, texture, shape, along with the high level spatial features of the image. These feature based extraction is prominently used in Content Based Image Retrieval whereby, desired images from a large image database on the basis of visual content that can be automatically extracted from the images themselves.

Image Understanding is the discipline that concerns the interpretation of an image and analysis of the image to give a decision about the image and the actions represented in it. Apart from the decision making process, it might include several sub processes like object identification, image retrieval and recognition of patterns based on their features.

1.1 Problem Statement

The problem involves extracting the various low level characteristic features of the image using the various techniques. These techniques are analysed to fetch images from a un-annotated image database, by a method known as content based image retrieval. These features extracted are used in the object recognition pattern recognition approach to further exemplify the utility of these feature extraction techniques. The problem involves understanding an input image to recognize patterns based on their features. The recognised patterns then generate a collective decision about the image as a whole and about the sub components present in the image. The Problem involves identifying the un-authorised terrorist air base camps set up at different locations.

1.2 Objective

This project aims at analysing an image and its components on the basis of feature extraction. The individual components are then combined together to generate a decision about the image. Further the extracted information about the individual components of the image is used for mainly two purposes viz:

- Object Based Recognition i.e. Object Identification
- Target Based Recognition i.e. Content Based Image Retrieval

1.3 Scope

- This project uses *Image Understanding*, which is the discipline that concerns the interpretation of an image and analysis of the image to give a decision about the image and the actions represented in it.
- This project uses *Decision Making Machine Learning* approach, to train the system to generate results about the area represented in the image.
- This project uses *Pattern Recognition*, which is the scientific discipline that concerns the description and classification of the patterns using *Feature Extraction.*
- The criteria used here is color, shape and texture for feature based extraction.
- Based on above criteria following *Image Retrieval* applications can be performed:

  Object Based Recognition

  Target Based Recognition

1.4 Product Functions

Image Understanding

- The individual components of the image identified by the pattern recognition module are used by used to fetch a desired result according to the decision tree based supervised machine learning approach.
- The project identifies the objects in the input image and also identifies the topological surrounding of the identified objects.
- The feature parameters based on the texture and shape characteristics of the image are used to identify the objects.

Pattern Recognition

- A feature is a cluster or a boundary/region of points that satisfy a set of pre-defined criteria .The criteria used here are color, shape and texture.
- This is divided into 2 sections:

  Object based

  Target Based
- Object and Target used for image retrieval can use any of the feature extraction techniques for comparison.

Database

- The user can import images to the database and delete images present in the database.

1.5. Organisation of the Dissertation

The dissertation provides a detailed overview of the system developed to analyse the various feature extraction techniques in the domain of image processing. It analyses the various techniques in their application to content based image retrieval and image understanding. Both of these applications are of upcoming importance to the field of pattern recognition and scene analysis, an area with critical applications of both medical and national importance.

The dissertation is organised as follows:

Chapter1 gives a brief introduction of the system. It describes the objective and the scope of the project. It also provides a brief overview of the problem statement handled and the various product functions developed in the project to accomplish the desired task at hand.

Chapter 2 provides a description of the domain and the various areas of importance to the project. It gives a detailed account of the various feature extraction techniques and their importance. It also describes the content based image retrieval approach and the various important phases of an image understanding approach. The decision tree machine learning approach used as a major backbone of the image understanding system is also stated in detail in chapter 2.

Chapter 3 describes the proposed system. It describes the system analysis and design of the system. The various algorithms are also described here to develop the various modules as described in the block diagram of the system.

Chapter 4 contains a brief of the various results of the system. These results analyse the feature based techniques and provide an insight to the application of feature based techniques in the image retrieval and understanding process.

The above stated modules of the dissertation are followed by the conclusion and the future research work of the project.

CHAPTER 2

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1 Image Processing

We are in the midst of a visually enchanting world, which manifests itself with a variety of forms and shapes, colors and textures, motion and tranquillity. The human perception has the capability to acquire, integrate, and interpret all this abundant visual information around us. It is challenging to impart such capabilities to a machine in order to interpret the visual information embedded in still images, graphics, and video or moving images in our sensory world. It is thus important to understand the techniques of storage, processing, transmission, recognition, and finally interpretation of such visual scenes. A two-dimensional image that is recorded by sensors is the mapping of the three-dimensional visual world. The captured two dimensional signals are sampled and quantized to yield digital images. Before describing the details of the digital image processing system, a brief overview of the digital images is presented.

2.1.1 Digital Images

A digital image is a representation of a two-dimensional image as a finite set of digital values. In image processing, the digitization process includes sampling and quantization of continuous data. The sampling process samples the intensity of the continuous-tone image, such as a monochrome, color or multi-spectrum image, at specific locations on a discrete grid. The grid defines the sampling resolution. The quantization process converts the continuous or analog values of intensity brightness into discrete data, which corresponds to the digital brightness value of each sample, ranging from black, through the grays, to white. A digitized sample is referred to as a picture element, or pixel. The digital image contains a fixed number of rows and columns of pixels. Pixels are like little tiles holding quantized values that represent the brightness at the points of the image. Pixels are parameterized by position, intensity and time. Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. Image is stored in numerical form which can be manipulated by a computer. A numerical image is divided into a matrix of pixels (picture elements).

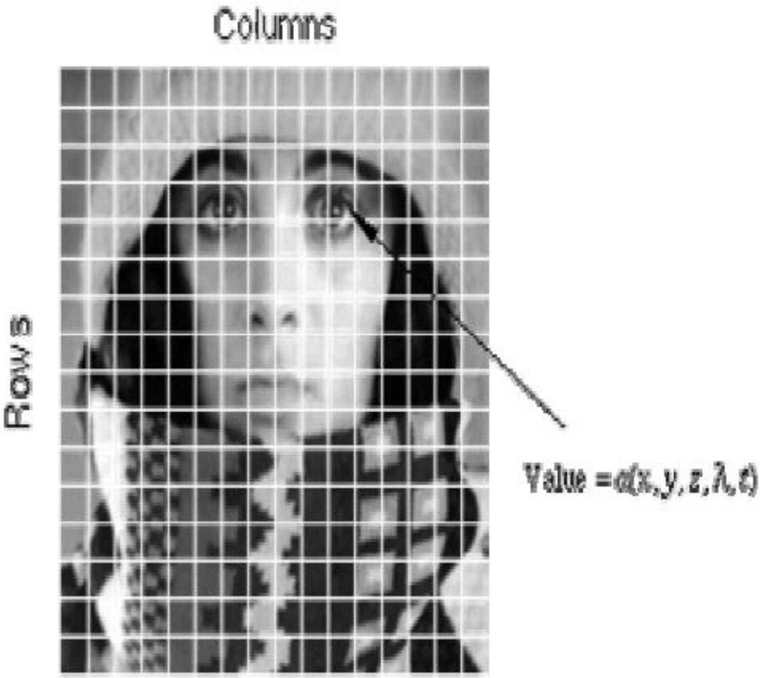


Figure 2.1: Digitization of a continuous image.

Digital image processing allows one to enhance image features of interest while attenuating detail irrelevant to a given application, and then extract useful information about the scene from the enhanced image. Images are produced by a variety of physical devices, including still and video cameras, x-ray devices, electron microscopes, radar, and ultrasound, and used for a variety of purposes, including entertainment, medical, business (e.g. documents), industrial, military, civil (e.g. traffic), security, and scientific. The goal in each case is for an observer, human or machine, to extract useful information about the scene being imaged.

2.1.2 Image processing operations

Image processing operations can be roughly divided into four major categories:

a) Image Restoration

b) Image Enhancement

c) Image Compression

d) Image Segmentation

Image Restoration: Restoration takes a corrupted image and attempts to recreate a clean image. As many sensors are subject to noise, they results in corrupted images that don't reflect the real world scene accurately and old photograph and film archives often show considerable damage.

Thus image restoration is important for two main applications:

a) Removing sensor noise,

b) Restoring old, archived film and images.

It is clearly explained in the Figure 2.2 and Figure 2.3.

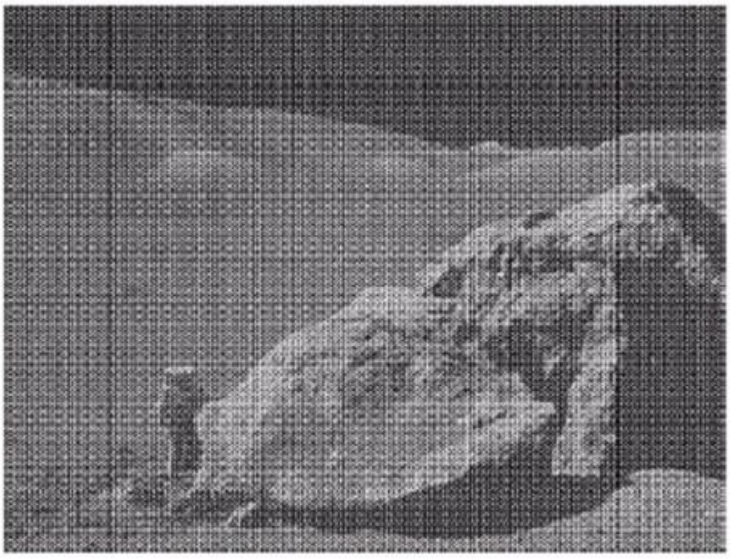


Figure 2.2: Original Image

Figure 2.3: Image after restoration

Image Enhancement: Image Enhancement alters an image to makes its meaning clearer to human observers. It is often used to increase the contrast in images that are substantially dark or light. Enhancement algorithms often play attention to humans' sensitivity to contrast.

Figure 2.4: Original Image before Image Enhancement

Figure 2.5: Image after enhancement

Image Compression: Image compression is the process that helps to represent image data with as few bits as possible by exploiting redundancies in the data while maintaining an appropriate level of quality for the user. The human eye has less spatial sensitivity to color than for luminance information. Large amounts of data are used to represent an image, so image has to be compressed when transferring from one place to another. Transform Coding has been a very popular technique for digital image compression. It is used in the

International Standard for Still Image Compression - JPEG (Joint Photographic Experts Group).

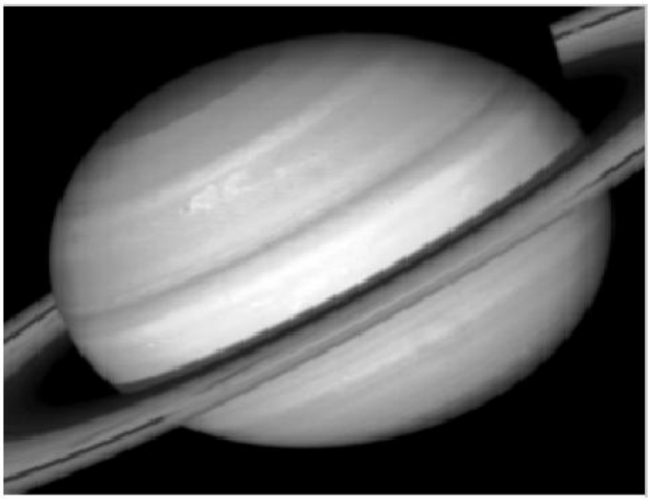


Figure 2.6: Original image before Image Compression

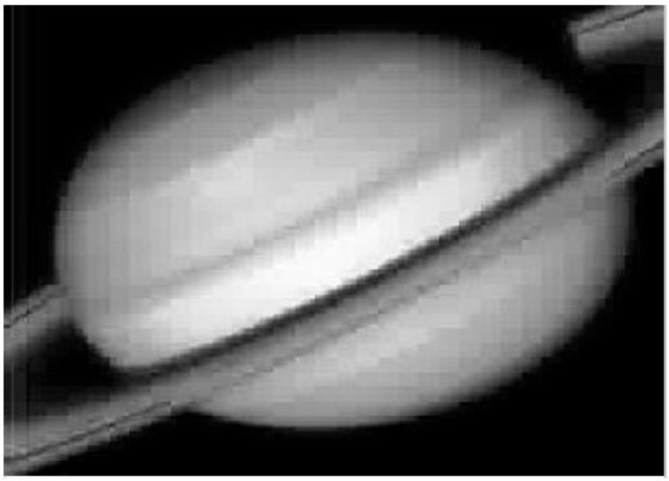


Figure 2.7: Image after compression

Image Segmentation: Segmentation is the process that subdivides an image into a number of uniformly homogeneous regions. Each homogeneous region is a constituent part or object in the entire scene. The objects on the land part of the scene need to be appropriately segmented and subsequently classified. Partitioning of an image is based on abrupt changes in gray level. If edges of the image can be extracted and linked, the region is described by the edge contour that contains it. The principal areas of interest within this category are the detection of edges of a digital image. An edge corresponds to local intensity discontinuities of an image. In the real world, the discontinuities reflect a rapid intensity change, such as the boundary between different regions, shadow boundaries, and abrupt changes in surface orientation and material properties. For example, edges represent the outline of a shape, the difference between the colors and pattern or texture. Therefore, edges can be used for boundary estimation and segmentation in scene understanding. They can also be used to find corresponding points in multiple images of the same scene. For instance, the fingerprint, human facial appearance and the body shape of an object are defined by edges in images. In a broad sense the term edge detection refers to the detection and localization of intensity discontinuities of these image properties. In a more restrictive sense, it only refers to localizations of significant change of intensity. Points of these localizations are called edges or edge elements. Edges are piecewise segmentation. They are both useful in computation of geometrical features such as shape or orientation. Edge detection is grounded on the assumption that physical 3-dimensional shapes in the scene, such as object boundaries and shadow boundaries, are clues for the characterization of the scene.

Figure 2.8: Original Image before Image Segmentation

Figure 2.9: Image after Segmentation

2.2 Image Understanding

When we look at an image, we can know some information without any other hints, for example, dark or light, etc. We can easily infer the different objects present in the image, the number of instances of objects of each type and their relationship with each other. That is because there exists a knowledge base in our experience and we can use the knowledge and extract the global information existing in an image. Moreover, we will take the image to be processed based on the knowledge, e.g., a dark image need to be enhanced in lightness, etc. However, such dependency on human thought process is not always reliable, especially in various critical applications of the scientific domain. Various image processing applications of medical, defence and other critical applications require image analysis and understanding operations. Image Understanding is a sub domain of the broader science of image processing. It involves analysing the image and generating an inference on the various aspects of the image, as per the application involved, a process known as **scene analysis**. The level of understanding in an image increases with the context the image and its data are linked together. Figure 2.10. represents the relationship between the context of the image data and the level of image understanding.

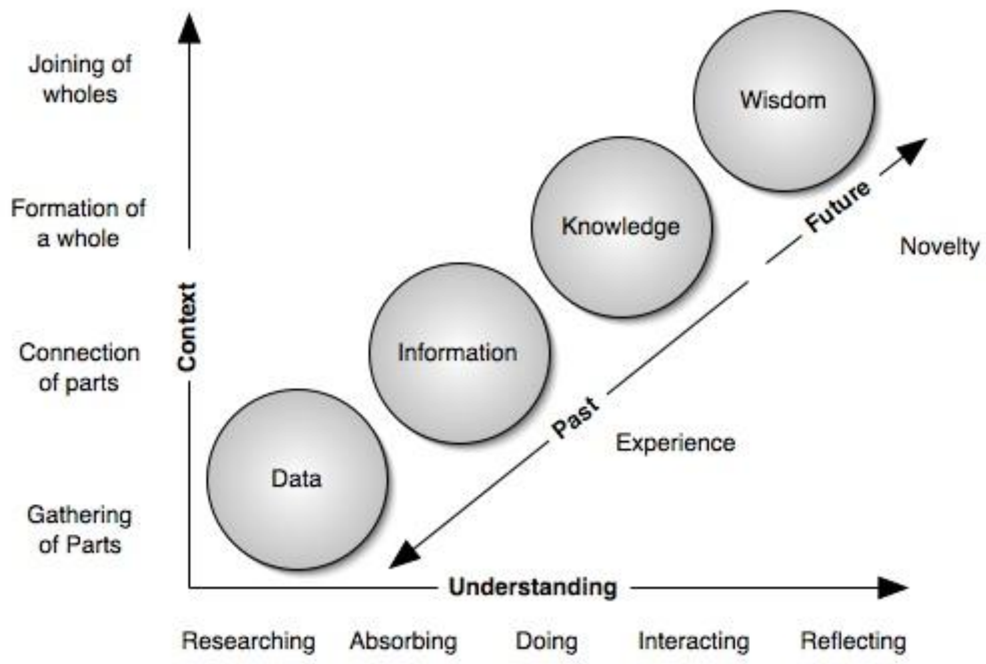


Figure 2.10: Relationship between image data and level of image understanding

The process of image understanding is used to predict the unknown i.e. the future; this would require the knowledge of the past. The understanding of the image is said to be at the researching when the parts of the image objects are gathered together to obtain data. This data when linked to each other through relationships leads to information which is absorbed by the system. The absorbed information is used to form the knowledge base through certain processes acting on it to refine the relationships of the data and generate the individual modules of the system. The knowledge generated can be used to reflect into the future and thereby join the entire system as a whole. This reflection into the future comes through the machine learning in the automated image understanding system. Thus, the level of description of the image data is closely related to the level of understanding of the system. Image understanding however requires both past experience and an insight to predict the future.

2.2.1: Steps involved in Image Understanding

The process of image understanding or scene analysis would require simulating the process of human intuitive thinking. Apart from the steps of image processing common to most systems, this process involves many processes which can be explained by human intuitive thinking, but always implemented by the numerical methods. For example, according to some transformations or statistics of an image, we want to find out the overall characteristics in an image, e.g., in histogram statistics, it can be seen that the image is biased to dark or light and by frequency transformation, the noise can be roughly predicted and filtered. Corresponding to each result, the images are mapped by a global feature and a linguistic meaning, just like human think. And, after the image improvement, a sub-global information of an image is considered to extract the object [28]. The object in an image can be easily recognized by human's intuition. That is, there is a knowledge base in the brain to know the structure of an object. The task of understanding analysed data is complex because it based on purely human processes, which comprise many different, important thought processes, including:

- Perception: differentiating, recognising, perceptual categorisation, orientation,
- Attention: unintentional attention, free attention,
- Memory: declarative, sensory, short-term, long-term (episodic, semantic, contextual), implicit memory,
- Thinking, calculating, abstracting,
- Forming concepts,
- Taking decisions, planning, predicting, problem solving,
- Understanding.

Human thinking processes leading to interpreting and describing complex data form the foundation for designing computer data analysis systems. The thought processes taking place in the human brain are used as the basis for designing a computer system which attempts to imitate natural processes and thus aims at the automatic, computer analysis of data. Each of the human thought process involved in the image analysis is simulated though the various image processing techniques to accomplish the task of setting an automatic image analysis system.

The various steps of an image processing system can be represented as in Figure 2.11.

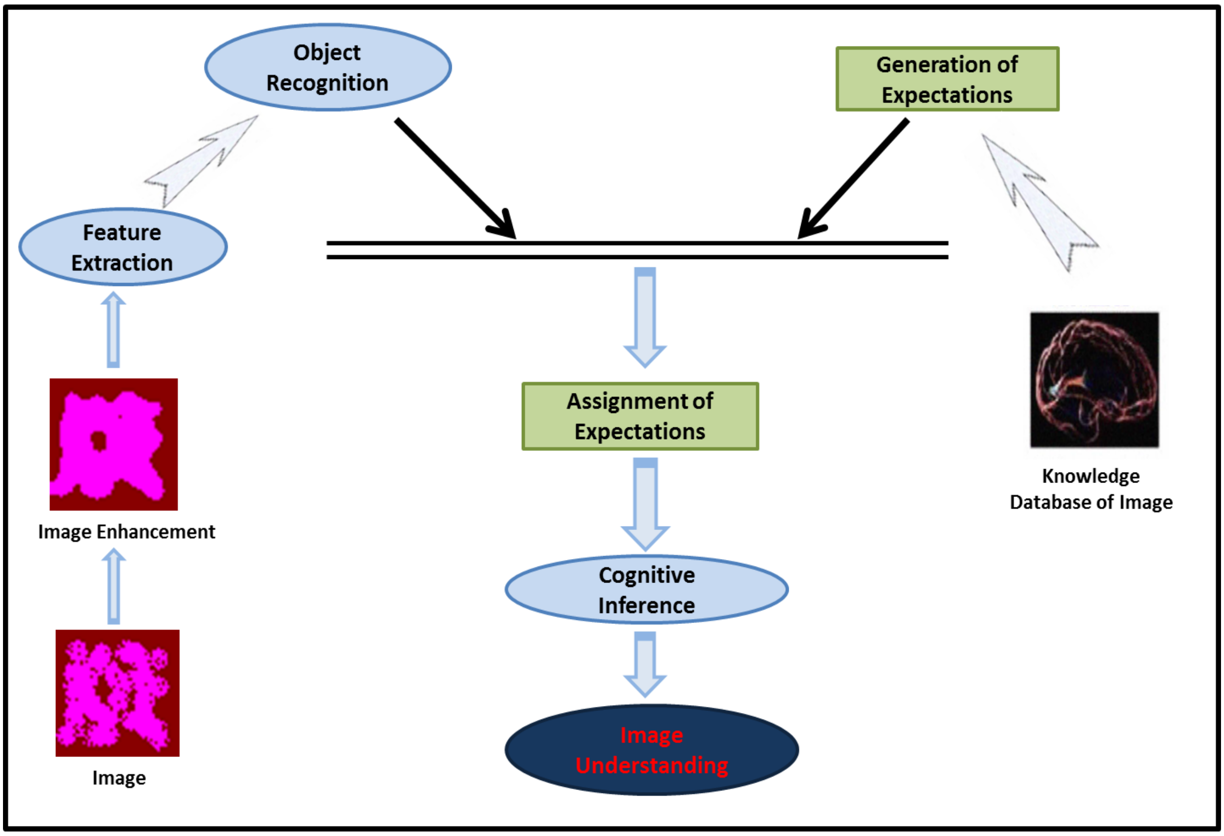


Figure 2.11: Image Understanding steps

The various steps in an image understanding system can be enumerated as follows:

- **Image Enhancement:** Image Enhancement process prepares the input image for the further image processing steps. It de-noises the image using Fourier Transform and other approaches. Image enhancement also resizes the image to make it suitable for the subsequent feature extraction techniques to be applied.

- **Feature Extraction:** Image understanding requires that the individual components of the image are represented by the features that best represent their properties. For example, a rocky terrain would be more suitably represented by the texture features rather than the color or shape features. Also, a clearly distinguishable object would have sharper and distinct shape features as compared to the color and texture features. Further discussion on the feature extraction techniques and their importance in extracting the individual properties and details about the image are further discussed in section 2.4

- **Object Recognition:** Object recognition stage of the image understanding process is used to extract the important objects in the image (scene). A relationship is then set up between the objects identified. The Object recognition is used to identify the important characteristic traits of the image and hence help in better descriptive analysis of the image (scene).

- **Generation of Expectations:** Based on the previous knowledge database and the research done from the existing knowledge, expectation values are generated. These expectations are used to ease the decision making process. For example, an image with bright clear sky and a bright sun would definitely represent an image of broad day light. Hence the expectation value of an object representing the sun would be higher towards representing a day and lower otherwise. Expectation values represent the probability of a result in case a particular object is recognised in the image.

- **Assignment of Expectations:** The expectation values generated according to the knowledge base are assigned to the objects recognised by the feature extraction phase on the input image. The various objects are assigned an expectation value.

- **Cognitive Inference:** Based on the expectation values of the individual objects in the image, the overall expectation value is assigned to the image. This expectation value is thereby used by the image understanding system to generate the final decision about the scene. The various techniques used to generate the result are based on machine learning (either supervised or unsupervised). They are based on the fact that the system is trained on a training set to generate an inference about the characteristic of the image. The various techniques which are used to generate this cognitive inference are presented in the section henceforth.

2.2.2 Image Understanding Techniques

The various techniques used in the cognitive inference in an image understanding system can be either supervised or unsupervised approaches to machine learning. The further sections in this topic describe few of the most commonly used Image Understanding Techniques.

2.2.2.1. Sparse Coding

Sparseness is a property of a random variable, such as the output of a linear filter when the input consists of natural images. Sparseness means that the random variable takes very small (absolute) values or very large values more often than a Gaussian random variable; to compensate, it takes values in between relatively more rarely. Thus the random variable is activated, i.e., significantly nonzero, only rarely. It is assumed here that the variable has zero mean. The probability density function p of a sparse variable, say s, is characterized by a large value (‘‘peak’’) at zero and relatively large values (‘‘heavy tails’’) far from zero. Here ‘‘relatively’’ means compared with a Gaussian distribution of the same variance. For example, the absolute value of a sparse random variable is often modelled as an exponential density. The exponential density is compared with the density of the absolute value of a Gaussian variable in Figure. 2.12. If the absolute value of a symmetric random variable has an exponential distribution, the distribution is called Laplacian.

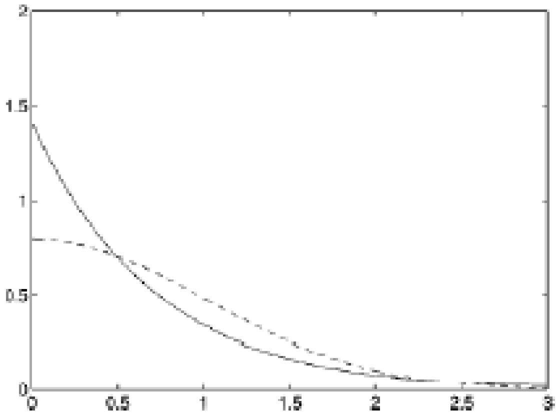


Figure 2.12: Illustration of a sparse probability density [20]. The vertical axis is the probability density, and the horizontal axis is the (absolute) value of random variable s. The sparse exponential density function is given by the solid curve. For comparison, the density of the absolute value of a Gaussian random variable of the same variance is given by the dashed curve.

The importance of sparseness lies in its ability to model the principal properties of simple cell receptive fields. Given natural image input, the outputs of linear filters that model simple cells are very sparse; in fact, they maximize typical measures of sparseness [20]. According to several authors an important characteristic of sensory processing in the brain is sparse coding. In sparse coding, a data vector x , e.g. an image (patch), is represented using a set of neurons so that only a small number of neurons is activated at the same time. Equivalently, this means that a given neuron is activated only rarely. In the simplest case, it is assumed that the coding is linear. For example, a widely used linear transform that performs sparse coding for many kinds of natural data is given by the wavelet transformation. One can also estimate a linear sparse coding transformation of the data by formulating a measure of sparseness of the components, and maximizing the measure in the set of linear transformations. In fact, sparsity is one form of non-gaussianity. Maximizing sparsity is thus one method of maximizing non-gaussianity. Therefore, at least in the case of linear coding, sparse coding leads to approximate dependence reduction of the components [21]. This was indeed one of the objectives of sparse coding in the first place. Thus, sparse coding can be considered as one method for ICA. At the same time, it gives a different interpretation of the goal of the coding method. The utility of sparse coding can be seen, for example, in such applications as compression and de-noising. In compression, since only a small subset of the components are nonzero for a given data point, one could code the data point efficiently by coding only those nonzero components. In de-noising, one could use some testing (thresholding) procedures to find out those components that are really active, and set to zero the other components, since their observations are probably almost purely noise [20].

2.2.2.2 Knowledge Based

The knowledge based cognitive inference system used in an image understanding system comprises of three knowledge bases. The first knowledge base acts as a pattern recogniser, extracting patterns/features from the image; the second knowledge base acts as a description builder, building the patterns/features into a primitive list of objects; and the third knowledge base acts as a knowledge retrieval and knowledge learning mechanism, matching the primitive list to stored facts in the memory of the knowledge-retention knowledge base. The knowledge based model incorporated two memory systems: Short Term Memory (STM) in the description builder and Long Term Memory (LTM) in the knowledge retender. STM is an immediate state of knowledge that is necessary for the functioning of a task and is used to declassify and expand internal description of items, and to compare, match and re-classify new data into an internal format. The generation of the rules for storing new information about an object is laid down through LTM using the same rules for extraction in STM and by extracting and regenerating information the model is able to flexibly manipulate information and re-classify this data as part of the image processing sequence. The transfer of information to LTM will only be activated in a condition of certainty, certainty being defined in terms of the Laws of Bayesian Logic. A general model for a knowledge based image understanding system is as described in Figure 2.13.

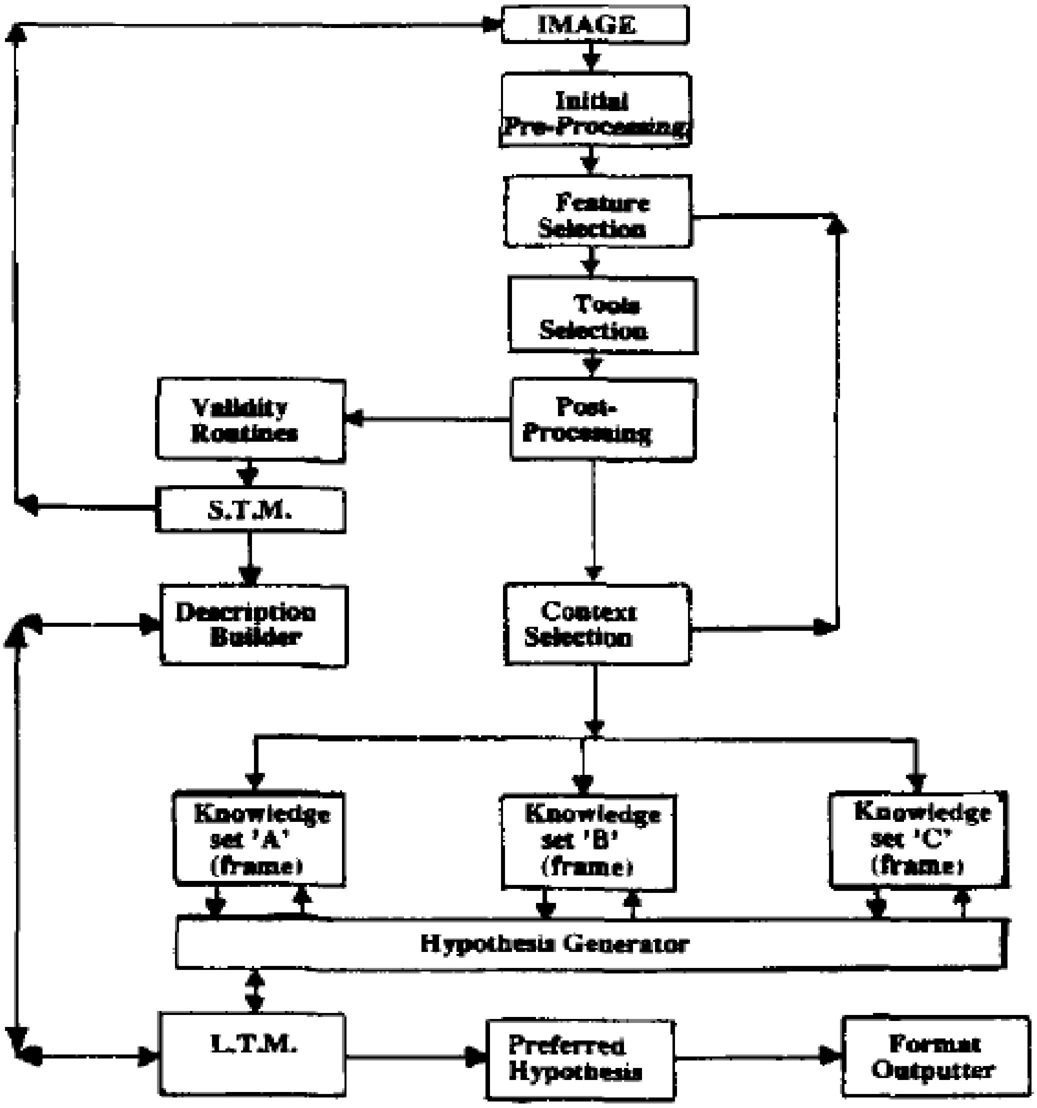


Figure 2.13: A general knowledge based image understanding system

The process of understanding is initiated via a preliminary gum as to what the image might be and the selection of tools to clarify the image is made on the basis of the initial assumption. The frame of reference under these circumstances might be labelled as 'POOR LIGHTING', 'BLURRED IMAGE' or 'COMPLEX IMAGE'. Stored under each frame would be a set of programming techniques best suited to resolving the characteristics of the image. A frame of reference is selected and initial assumptions are made about the type of image the system is dealing with, and this initial frame of reference is maintained until such a time as the evidence for or against the hypothesis is sufficient to alter the Context. The features selected for analysis depend on the image frame and a fixed sequence of processing could be generated a possibly a parallel process developed to cope with the necessary decisions. A typical set of features might consist of colour, texture, brightness, lines identification, line orientation, image depth etc.

Based on characteristics of the image and the tools selected, the system assigns a resolvability factor to the image. In terms of the problem space this means that, the more complex the search or the lower the resolvability factor, the greater the degree of rule application, and the more the need to gather evidence to prove or disprove the initial hypothesis. This is part of the post-processing stage of the image, and the results are verified and then stored in STM. A context selector is employed to route the description builder through the correct set(s) of knowledge and a hypothesis generator functions as the "brain" of the system, presenting the domain specific knowledge to the description builder. The job of the context selector is critical to the success of the system because context can be used to restrict the size of the problem domain by reducing the number of possible interpretations of a Scene. The link between the description builder and the hypothesis generator operates at the level of a categorisation mechanism. Sorting and labelling the information about the hypothesised scene into meaningful wholes. Input to the description builder comes from the post-processor via validated output from STM. If an adequate description cannot be built from the available knowledge sets then the image would be re-analysed at the initial pre-processing stage. In similar circumstances, where the context selector is unable to tap adequate information from the knowledge base, the system re-selects features from the Scene to generate a different post-processed image [25]. In the event of an adequate image being generated by the description builder, the context selector is finely tuned to acquire the most pertinent knowledge set.
Following this output from the description builder is directed to LTM and then onto the preferred hypothesis sequence. LTM is used as a part of the mechanism for adding new facts to the database following the decomposition of the description by the hypothesis generator. The new fact would then be automatically added to the pre-contextualised knowledge *s*. The bottom-up processes of the system constitute a direct route from the critical pre-processing through feature selection, to tool selection and finally post processing. The top-down processing consists of a re-processing of the image based on an inadequate description building sequence, and a re-selection of features at a secondary level in the event of a contextual mismatch. The mobilisation of knowledge sets is an independent process, consisting of the evolution of hypotheses a NIC based generation, and the setting of depth and

breadth searches for process limit specification. Output to the description builder consists of a pre-described sequence of operations upon which a functional description can be built for ascribing meaning to the post-processed image. The final stage of the model is the output and acquisition of knowledge for the scene. with the re-composition of description by the hypothesis generator via LTM, and the output of the preferred hypothesis to the formatter for communicating to the user.

2.2.2.3 Fuzzy Logic Based

A fuzzy logic based cognitive inference system based image understanding is based on fuzzy grammar and image object recombination [11]. An example of digital image scene analysis is presented in this section. Figure 2.14 represents some digital image scenes.

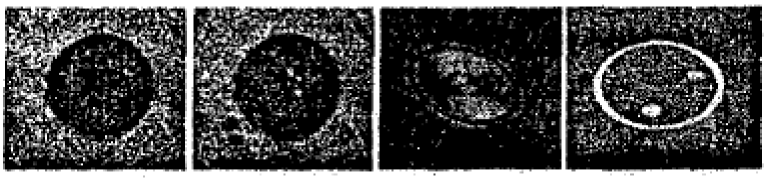


Figure 2.14: Examples of image scenes

The above described Figures represent grass fields where a surface was modified by an unknown factor. Mentioned scenes are described by simple rules. It includes of up to four circles disposed over the field. The following four classes are described by their characteristic features, single circle, big circle with two small circles positioned out of the big one, radial circles and big circle with two small circles included. Those four classes are easy to describe by predicates [11].

According to this classification, steps of scene analysis are the following.

Alphabet definition: Alphabet of the formal system consists of all objects (simple and recombined ones) expected to appear on the specific image scene. In this example it may be:

A = {*GraphicElt* [*Dot* ,...],

GraphicElt [*Circle, ...*]}.

GraphicElt[*Line* ,...]},

GraphicElt[*Curve*, ..], . . .}

Grammar definition: Grammar definition includes transformation rules description. A list of all rules providing recombination dots to lines, lines to curves, etc. is too long. An example would be:

RewritingRule [*DotsToCurve, {Dot, Dot},{Curve}*].

Object recombination: Acquired dots are recombined to higher level objects applying formal language transformation rules. Lines are and curves found at the next level of recombination. A final level of objects consists of circles, crosses, some curves, lines and dots not included in the process of recombination creating higher level objects. Examples of objects constructed are *GruphicElt[ClassC2, ...*] and

GraphicElt[ClassNone, ...], where first object represents the image second class and the second one not classified image.

Scene analysis: At this stage, a set of final level objects is analysed applying formal theory predicates. Predicates describing four above described classes of images are used to identify whether the given image represents a scene of some of four classes or it cannot be classified to any of mentioned classes.

Fuzzy Labelling and Inference: The features like "CIRCLE", "SQUARE", etc., are applied to define the membership grade of the shapes in the image. And, the labelled shapes and information are used to infer the fuzzy rule to decide the real objects and extract them using the rule like "if the object contains CIRCLE above LONG Bar, then it is a Tree".

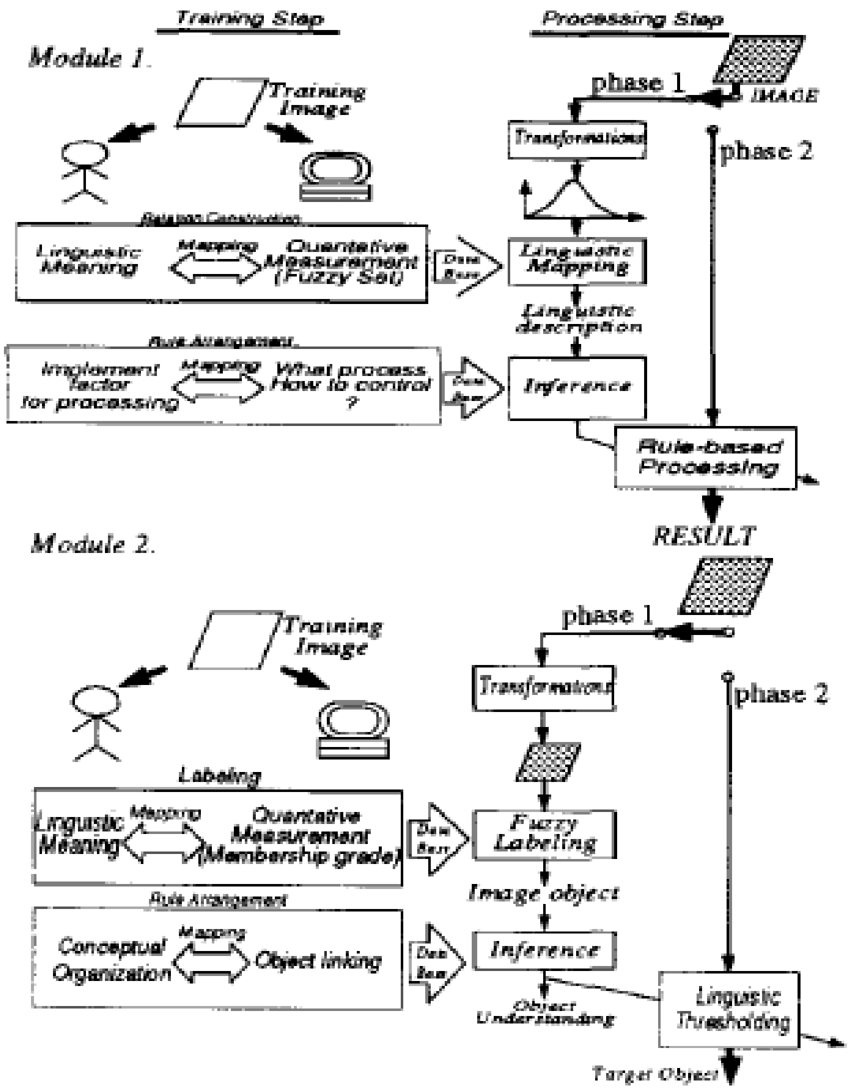


Figure 2.15: Fuzzy logic based image understanding system[14]

2.2.2.4. Neural Network Based

Neural network is a vital machine learning method. The neural network can however be used in two different ways. One is using neural network's auto-associative memory function, and the other is by the function of hetero-associative memory. If hetero-associative memory function is used, for the network stored pair, one is the image pattern itself and the other is the class to which the image belongs [15]. If auto-associative memory is used, the two elements of the stored pair are images themselves. When one image is fed as input into the system, maybe there some differences between the inputted image and the network stored standard image due to the distortion, noise, some part missing or other factors, the correct recognition result is obtained after the network operation because of the network's high robustness and fault-tolerance . If the system uses the auto-associative memory function, the standard image can be recalled (associated) after the network convergence. By comparison with the standard image, it can be known which classes the inputted image belongs to. If system uses the hetero-associative memory function, it can directly be known which classes the inputted image belongs to after the network convergence. There are two types of neural network which have heteroassociative memory function, i.e., Kosko's Bidirectional Associative Memory Model ' " and Multilayer Feedforward Model. Autoassociative memory is a special form of heteroassociative memory. Besides these two types of heteroassociative memory network, there are many other networks which can do autoassociative memory, such as M-P model, Hopfield model, Loop Neural Network Model, Nonlinear Continuous Neural Network Model, etc.

2.2.2.5. Cognitive Approach

In the cognitive data analysis process, the process whereby the system learns new solutions which may impact the decision-making solution obtained is of key importance. So far, in the cognitive categorisation processes, the understanding of analysed data was based on the classical cognitive analysis process, whereby connections were identified between pairs of consistent expectations of the system acquired from expert knowledge bases and characteristic features extracted from analysed datasets, and this led to cognitive resonance during which the above connections were determined as consistent or inconsistent. Only pairs that were completely consistent were selected for further analysis and a group of solutions could be defined which included the identified consistent pairs [28]. This definition of the group made it possible not only to recognise the analysed data by naming it correctly, but also made the understanding of data complete, which lead to determining semantic features of the analysed data. Conducted research has shown that in this solution, pairs of characteristic features of the analysed data and expectations generated based on the expert knowledge base collected in the system which were not consistent were omitted at further analysis stages. This made it possible to envisage a situation in which the system encounters a solution it does not know and which is not defined at all in its bases. The question is, is it possible to recognise this type of a situation? Yes, the solution proposed in this publication shows that it is possible to introduce a stage at which the system is trained in solutions new to the system. This process is possible only when the set of solutions obtained (both optimum ones and those eliminated from further analysis) is used to create a set of features of analysed data and a set of new expectations not defined in the original bases of the system. The new features and expectations are input into the system base in which data is re-analysed, this time using the much broader expert knowledge set containing new patterns learned by the system. Such patterns constitute an extended expert knowledge base which the system uses to generate a set of expectations, and these are compared to the set of characteristic features of the analysed data. This process thus becomes an enhanced process of cognitive analysis based on cognitive resonance for learning systems. A system can be trained in any situation and this training can be multiplied depending on the needs and the necessity of extending the knowledge bases built into the system.

2.2.2.6 Decision Tree

Decision trees can be described by following characteristics

- Internal node represents an attribute.
- Branch represents a value of an attribute.
- Leaf node represents a class of a classifying attribute.
- Decision tree is traversed from top to bottom by performing test on each internal node that comes in the way until the leaf node is encountered.
- If attribute is continuous rather than discrete then a threshold is formed and tests are performed on that threshold value.
- Leaf node contains records belonging to the same class.
- A sample decision looks like:

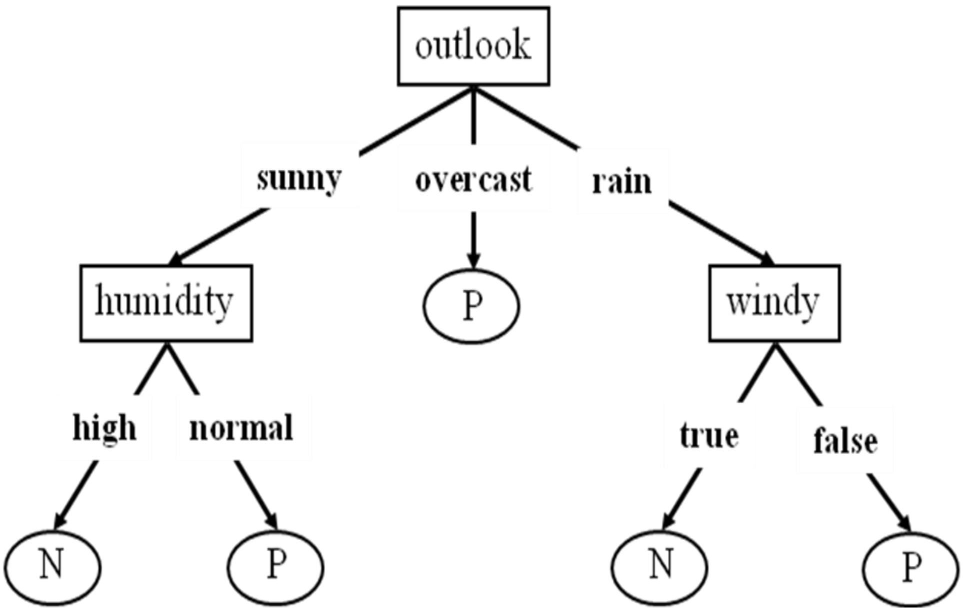


Figure 2.16: Sample Decision Tree

Example: Whether to go for play or not?

- a set of attributes and their possible values:
  - outlook sunny, overcast, rain
  - temperature cool, mild, hot
  - humidity high, normal
  - windy true, false

Each instance in the training set is associated with a class. "N" stands for not play and "Y" stands for play.

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Figure 2.17: Sample Training Data

2.2.2.6.1 Using Decision Trees for Classification

Example:

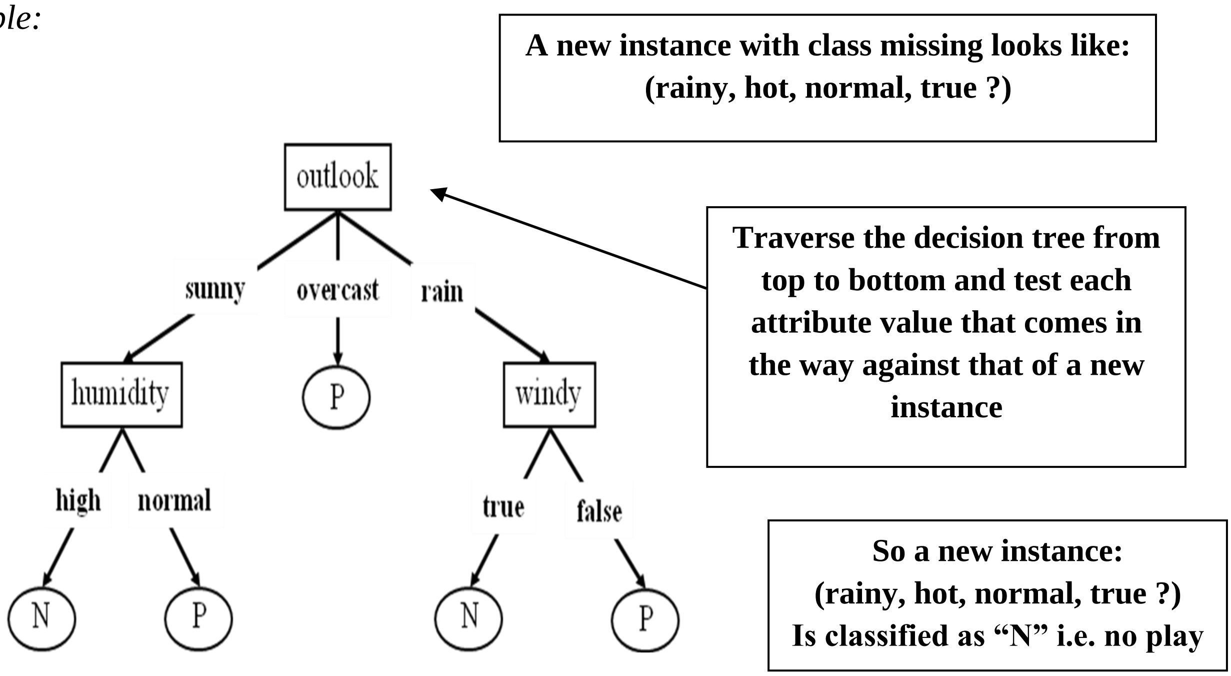


Figure 2.18: Example-Using Decision Tree for classification

2.2.2.6.2 Types of Decision Trees

There are two types of Decision Trees:

- *Classification tree* is used to predict a class for a new instance
- *Regression tree* is used to predict a real number.

2.2.2.6.3 Advantages of Decision Trees

Amongst other decision methods, decision trees have various advantages:

- Decision trees have the ability to handle discrete attributes as well as continuous attributes.
- Unlike other classification techniques which require data normalization, removal of missing attribute values etc., they just require a simple data preparation.
- Decision trees are very easy to understand and visualize.
- Decision tree model can easily be tested and validated. In other words statistical test cases can easily be applied on the decision tree model to check the reliability of the model.

- Decision trees have separate to handle missing attributes.
- Decision trees work well for large amount of data. Large amount of data can be analysed in lesser time. They are known for their good performance against larger data sets.

2.2.2.6.4 Limitations of Decision Trees

- Practical decision-tree learning is based on greedy algorithm where locally optimal decisions are made at each level of the tree. Such algorithms cannot guarantee to return the globally optimal solutions. That is why the problem of learning with Decision tree is NP Completes.
- A tree generated may over-fit the training examples due to noise or too small a set of training data. This is called over fitting in classification.

2.3 Image Retrieval

In this fast application based system where huge amount of data is handled at runtime, fast image retrieval is required. The following sub section describes the traditional image retrieval where the image data was annotated. But due to the rapid demand of an image retrieval system and the non-availability of annotated image database, content based image retrieval system has gained importance. The content based image retrieval system is also described later in this section.

2.3.1 Traditional Image Retrieval System

Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to users' interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines. Early techniques were not generally based on visual features but on the textual annotation of images. In other words, images were first annotated with text and then searched using a text-based approach from traditional database management systems.

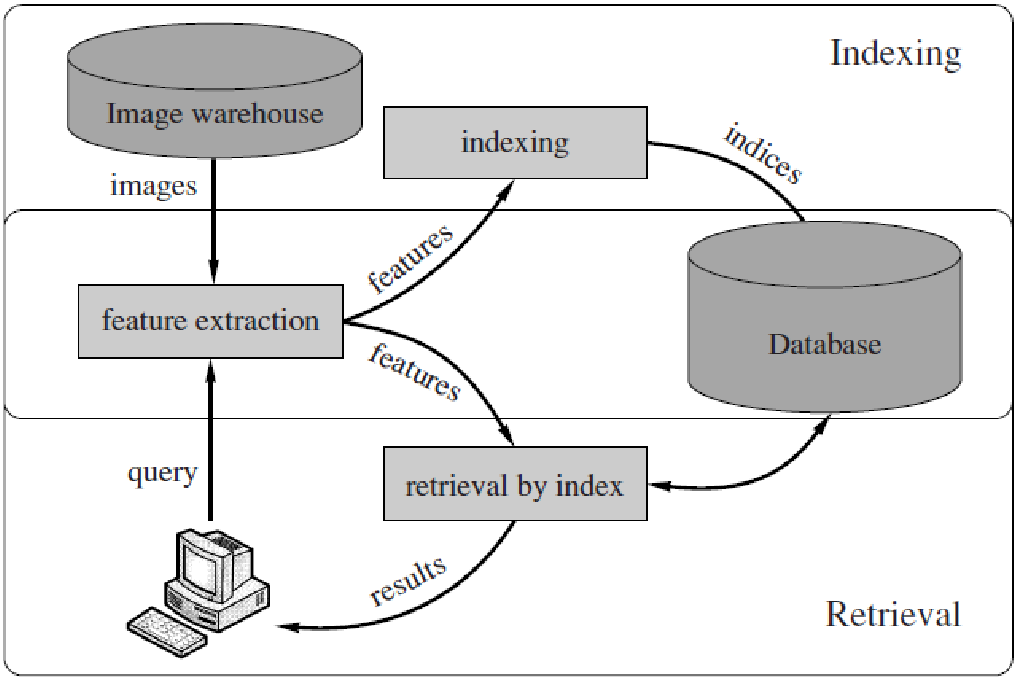


Figure 2.19: Traditional architecture of image retrieval system [17]

Text-based image retrieval uses traditional database techniques to manage images. Through text descriptions, images can be organized by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard Boolean queries. However, due to rapid growth of technology the abundance of digital images a need for a system to organize un-annotated data has risen. Content Based Image Retrieval System addresses the problems of the early methods of handling huge critical database. It organises and analyses the data on the basis of the visual content of the images. A typical architecture of a traditional information (image) retrieval system is as shown in Figure 2.19.

The traditional image retrieval system had two basic functional modules: indexing module and the retrieval module. The indexing module generates indexes through a hash function and saves the indexes in the database for faster retrieval of images from the database. The retrieval module is responsible for processing the input query image and generating the corresponding result from the image database.

2.3.2 Difficulties of Content Based Image Retrieval System

The revolutionary internet and digital technologies have imposed a need to have a system to organize abundantly available digital images for easy categorization and retrieval. The need to have a versatile and general purpose content based image retrieval (CBIR) system for a very large image database has attracted focus of many researchers of information-technology-giants and leading academic institutions for development of CBIR techniques. Major difficulties faced [17] during content based retrieval are as follows:

- **Semantic Gap**:

  Semantic gap refers to the difference in the way a human perceives an image and the way a system perceives it on the basis of its certain characteristics. A major problem in content based retrieval is of semantic gap. A human compares two images, based on their semantics, whereas a retrieval system relies on comparison of feature vectors corresponding to visual image features.

- **Inconvenience of Querying-by-Example**

  One of the most frequently used kinds of queries in content-based image retrieval systems is a query-by-example. Such a query is not convenient for the user, who may not have such an example; moreover, the user addresses the retrieval system in order to find such an image. Many systems suggest the user to select the image to be used as a query from a random set of images taken from a database. Such a random set may not contain an appropriate image, and the user is obliged to look through one set after another trying to find a desired query. Users prefer text queries, since it is simpler to describe content of the desired image in words. However, in order to search by a text query, the system should possess some text information on images.

- **Data Volume and Retrieval Performance**

  Images can be retrieved thorough the indexes attached to them in the form of text or a hashing value generated by a hash function. However in case of un-annotated image database this form of *textual retrieval* becomes difficult, content based retrieval needs to be done in this case. Content based image retrieval is the retrieval of images from a large database depending on the visual content of the images rather than the textual content associated with the images. This visual content of the image can be analyzed through a feature vector of a specific dimension, depending on the feature extraction technique adopted for retrieval. The greater, the dimension of the feature vector, the greater is the system's performance. However the increased data space affects the processing speed and the data management of the system.

2.3.3 General Content Based Image Retrieval System

Content based image retrieval is the retrieval of images from a large database depending on the visual content of the images rather than the textual content associated with the images.
This visual content may be the *general visual content*, represented by the low level feature descriptors like color, texture, shape, spatial relationship. The visual content may be *domain specific visual content*, specific to the application the CBIR system is developed for, for example the fingerprint matching system. The Domain specific content is application dependent and requires specific domain knowledge. A typical Content Based Image Retrieval System is generally represented as in Figure 2.20.

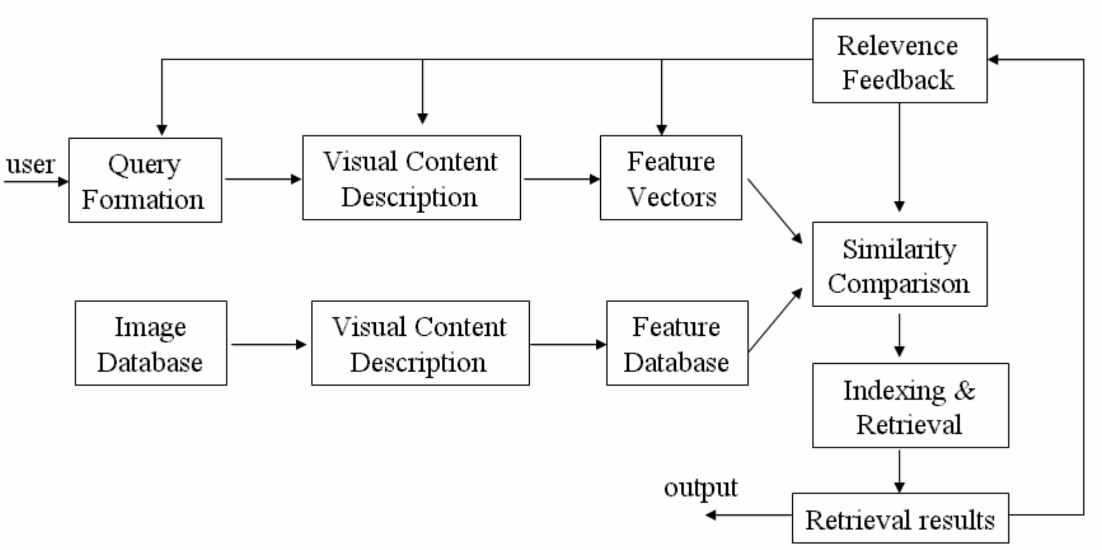


Figure 2.20: A Typical Content Based Image Retrieval System

The various functional blocks of a general Content Based Image Retrieval System include an Image Handler, a Feature Extractor, a Similarity Comparator, a Database Handler and a GUI handler amongst its major components. The *Image Handler* module analyses the image format of the input image. It reads the input image and converts the image into a data matrix which stores the color components and intensity values according to the different color spaces for each pixel. If RGB color space is considered for example, then the data matrix would be a 3 dimensional matrix of the dimension; "*Height X Width X 3*", storing the R, G and B component values for each pixel. The Image handler module is also responsible for converting the data matrix back to the image file while saving the results generated by the CBIR system, i.e. while saving the generated result in the form of an image rather than a matrix. The *GUI handler* module manages the entire GUI of the system required to ease the end user in specifying the query and to view the results. The *Feature Extractor* module is responsible for extracting the various low level and high level features of the image using the various techniques for extracting the color, texture and shape parameters of the image. The feature extractor would calculate the features of the image as a whole and of various sub images within the main image. The features are then saved in the database in the form of tuples of the different tables. The feature vector comprises of the pixel coordinates of the top left corner of the sub image along with the feature parameters corresponding to the method used for feature extraction.

A typical feature vector looks like,

$$F_i = \{ x_i, y_i, f_1, f_2, \ldots f_n \}$$

Where, F_i is the feature vector corresponding to the i^{th} sub image. x_i and y_i are the pixel coordinates of the top left corner of the rectangular sub image. And f_1, f_2,....f_n are the parameters generated by the various techniques. The feature vectors corresponding to all the sub images or the image as a whole are stored in the database for future retrieval by the Similarity Comparator module. Further details of the Feature Extractor and the various techniques are discussed in subsequent sections. The storing of these feature parameters and their retrieval is done from the database by the Database Handler module. The calculation of the image parameters is a one-time task thereby reducing the running time and the response time of the system required to generate the results. The system does the task of understanding the input query and calculating the similarity between the input query image and the images or the sub images in the database. This task is done by the *Similarity Comparator* module of the system. The similarity may be measured using the various distance measuring techniques. These are general modules of a CBIR system. Other module can be added to the system depending on the application the system is being developed for.

2.4 Feature Extraction

Feature extraction algorithms and similarity measures used for image comparison underlie any CBIR system. A CBIR system uses the visual contents of an image such as *color*, *shape*, *texture*, and *spatial layout t*o represent and index the image. In a typical content based image retrieval system (Figure. 2.20), the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors. The similarities / distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme.

All content-based retrieval methods can be classified into classes depending on the features they use: search by color, texture, or shape. Each class, in turn, is divided into subclasses by the type of the algorithm used for constructing the feature vector. Some researchers classify spatial features of images into a separate class. Spatial features are those reflecting spatial layout of homogeneous image regions in terms of one or another feature: for example, region of the same color or the same texture or a particular object. In other words, these are features of one of the classes (color, texture, or shape) with additional information on spatial layout. In what follows, the common algorithms for color, texture, and shape feature extraction are considered. For each of these classes, a more detailed classification is presented in a separate section.

2.5 Color

Color feature is the most significantly used visual content descriptor. It plays a vital role in searching collections of color images of arbitrary subject matter. Color has a strong significance in the human visual perception mechanism. Besides the simplicity involved in analyzing the color components of any image, color also offers special properties of being easy to extract and analyze and its property of invariance with respect to the size of the image and orientation of objects on it. These characteristics of the color component of an image make it the most sought after choice in the image retrieval systems. The quality of color feature vector greatly depends on the color space selection. The different color spaces behave differently with different applications and algorithms applied.

2.5.1 Color Spaces

A color space (also referred to as a *color model* or *color system*) is a specification of a coordinate system and a subspace within this system where each color is represented by a single point. Thus, each color in a color space has its *color coordinates.* [17].*C*olor model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or *color components.* The commonly used color spaces are as follows:

RGB: red, green and blue. The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. RGB is a *device-dependent* color model: different devices detect or reproduce a given RGB value differently. It is widely used for image display in monitors, cameras, etc. due to the close device dependence of the RGB color model.

CMY: cyan, magenta, yellow or the CMYK: cyan, magenta, yellow and black are two other similar color models. The CMYK color model is a subtractive color model, used in color printing since the color printing process also involves 3 inks and uses subtractive process to generate the different colors. The conversion between RGB color space and CMY/CMYK color space is rather a little difficult since both the color spaces are device dependent. Hence separate color management systems need to be employed to interchange between the two color spaces.

HSV: hue, saturation, value (also referred to as HSB, B for brightness) and HSI (I for Intensity) are two other most commonly used color space models used in image processing and analysis applications. It is a cylindrical coordinate representation of the RGB points in the Cartesian coordinate representation of the RGB color space. The hue is invariant to the changes in illumination and camera direction and hence more suited to object retrieval. HSV color model is a more intuitive way of describing the color parameters in comparison to the other color models. It represents the way how human eye and brain observes and analyzes the color in terms of its chromaticity and brightness instead of realising a color image as a sum of 3 primary sub images. The following expressions convert a RGB color space system in the range [0,1] to a HSV color space in the range [0,1]:

$$V = \max(R,G,B)$$

$$S = \frac{V - \min(R,G,B)}{V}$$

$$H = \frac{G-B}{6S} \text{, if } V = R$$

$$H = \frac{1}{3} + \frac{B-R}{6S} \text{, if } V = G$$

$$H = \frac{2}{3} + \frac{R-G}{6S} \text{, if } V = B$$

The traditional color space for representing digital images is RGB color space. However image processing algorithms typically use HSV color space.

The following sections describe the various color feature extraction techniques.

2.5.2 Color Based Feature Extraction Techniques

Color represents an important feature in images. These color features are easily represented by the various moments and by the histogram representation. In the following sub section the various techniques for extracting the color based features are represented.

2.5.2.1 Color Moments

Color moments have been successfully used in many retrieval systems (like *QBIC*), especially when the image contains just the object. The *first order* (*mean*), the *second* (*variance*) and the *third order* (*skewness*) color moments have been proved to be efficient and effective in representing color distributions of images.

Mathematically, the first three moments are defined as:

Energy = $E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij}$

Standard Deviation = $\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^2\right)^{\frac{1}{2}}$

Skewness = $s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^3\right)^{\frac{1}{3}}$

The CBIR systems usually use 3 numbers (parameters) for each band of the image. Hence a total of 9 parameters (in case of 3 bands: RGB, HSV) are used to represent the entire image. The Color moments are easy to compute and compact set of parameters facilitating easy similarity distance measurement.

2.5.2.2 Color Moments with Fuzzy Regions

Color moments though a very simple and efficient technique has a few shortcomings. It does not take into account spatial layout of colors [22][17]. To overcome this, a modification to the above technique is practised. The image is partitioned into layout regions of fixed size (or more complicated image segmentation) and calculating features of color distribution for each of them. The image is divided into “fuzzy regions.” The following five regions are introduced: central ellipsoidal region and four surrounding regions as in Figure 2.21. All regions are defined by membership value described by a membership matrix. An example of a membership matrix is shown in Figure 2.21.

According to this membership matrix, pixels located strictly in the centre of the image completely belong to the central region and thus affect the feature vector of the central region only. The closer the pixel to the region border, the lesser is its influence to the region’s feature vector. Pixels located on a border separating two regions affect the feature vectors of both regions. Experiments show that such an approach makes it possible to improve retrieval results in the case of more complicated queries, when it is required to take into account spatial layout of objects on the image.

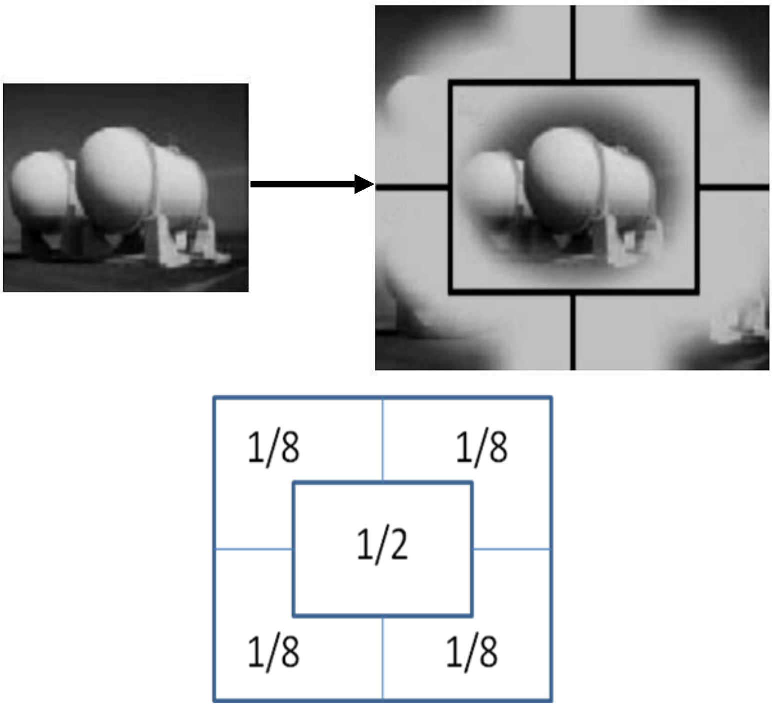


Figure 2.21: Partitioning image into fuzzy regions, Membership matrix

The moment equations described in above section are applied to each fuzzy region to generate 9 parameters (for 3 bands) for each fuzzy region.

The overall Parameters of the image are then computed as

$$F(v) = \sum_{i=1}^{5} \mu_i * f_i(v)$$

Where, $F(v)$ is the value of the overall parameter for parameter v (v = mean, standard deviation, skewness) μ_i is the value membership value of a fuzzy region in an image and $f_i(v)$ is the value of the parameter v in the i^{th} region. Here though each of the five regions has 9 parameters but the overall parameters of the image remain 9 only. Color moments with fuzzy regions though computationally expensive as compared to the Classical Color Moments approach, generate more efficient results due to larger contribution by the central pixels than those at the boundary.

2.5.2.3 Color Histogram

The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing both the distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle. Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in RGB space, or hue, saturation, and value in HSV space), a *histogram*, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases. Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance in many applications. In addition, color histogram does not take the spatial information of pixels into consideration, thus very different images can have similar color distributions. This problem becomes especially acute for large scale databases. To increase discrimination power, several improvements have been proposed to incorporate spatial information. A simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. As introduced above, the division can be as simple as a rectangular partition, or as complex as a region or even object segmentation. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

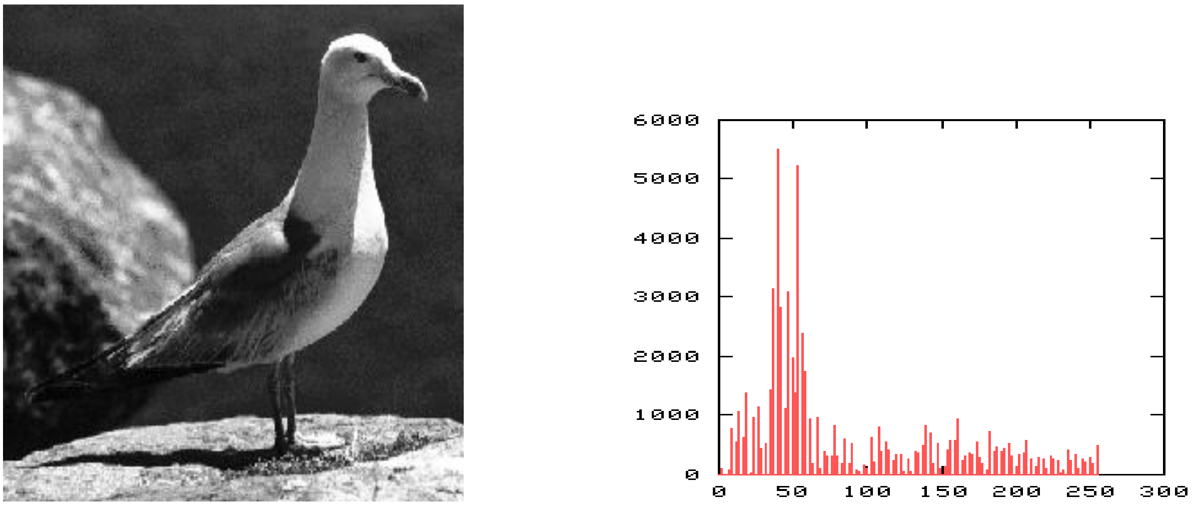


Figure 2.22: Image with an Example Histogram

2.6 Texture

Like color, image texture is also very important for visual human perception. It gives us information on structural arrangement of surfaces and objects on the image. Texture is not defined for a separate pixel; it depends on the distribution of intensity over the image. Texture possesses periodicity and scalability properties; it can be described by main directions, contrast, and sharpness[18]. Texture analysis plays an important role in comparison of images supplementing the color feature. There exist many methods for representation of information on image texture and comparison of images on the basis of texture [17]. Texture features are classified into statistical, geometrical, model, and spectral ones. Statistical features describe distribution of intensity over the image by means of various statistical parameters. The most frequently used statistical features include:

- general statistical parameters calculated from pixels' intensity values
- parameters calculated based on the co-occurrence matrices
- texture histograms built upon the Tamura features

It is common for geometrical features to describe texture with *texture elements*, or *primitives.* They specify texture by defining texture primitives and rules of their placement relative to one another. Structural methods are not appropriate for describing irregular textures. Model methods of texture analysis rely on construction of a model that can be used not only for description but also for texture generation [26][3]. Spectral features describe texture in the frequency domain. They are based on expanding a signal in terms of basis functions and using the expansion coefficients as elements of the feature vector. Few of the frequently used techniques for extracting texture features are as described in the following section.

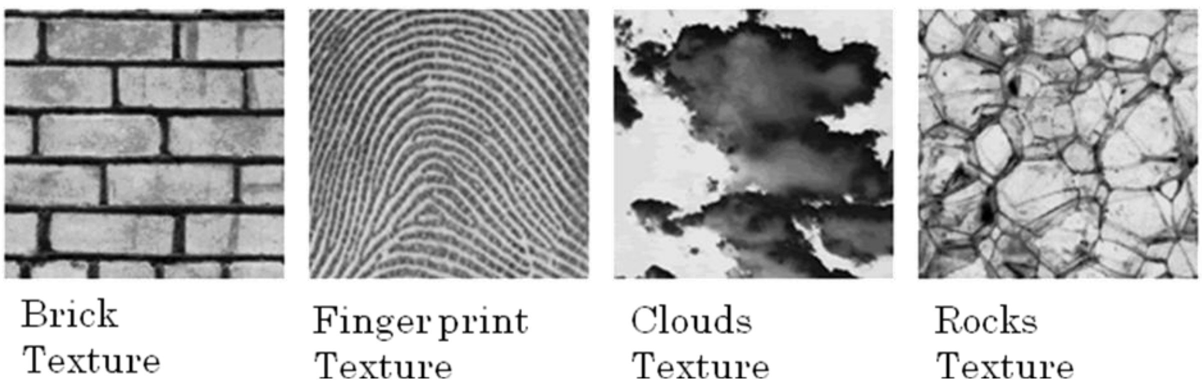


Figure 2.23: Few examples of different textures

2.6.1 Texture Based Feature Extraction Techniques

The various textures in an image are captured by the various techniques like grey level co-occurrence matrix, Gabor filter and wavelet filter. Few of these techniques have been described in the sub section henceforth.

2.6.1.1 Gray Level Co-Occurrence Matrix

The grey level co-occurrence matrix is defined by Haralick et al. as a frequency matrix of pairs of pixels of certain intensity levels located with respect to one another in a certain way. Mathematically, the co-occurrence matrix C of an image I of size $N \times M$ is defined as follows:

$$C(i, j) = \sum_{p=1}^{N} \sum_{q=1}^{M} \begin{cases} 1, \text{ if } I(p, q) = i, \\ I(p + \Delta x, q + \Delta y) = j, \\ 0, \text{ otherwise,} \end{cases}$$

Where, $(\Delta x, \Delta y)$ is an offset (or displacement) vector determining mutual location of pixels and $I(p, q)$ is intensity of a pixel located in position (p, q). An example of a co-occurrence matrix for an image region for $(\Delta x, \Delta y) = (1, 0)$ is shown in Figure 2.24 [17]. It is not difficult to see that the offset $(\Delta x, \Delta y)$ makes such a representation of the texture feature sensitive to image rotation, which is usually undesirable for a CBIR system. Therefore, it was suggested to construct several matrices with different offset vectors corresponding to image rotation through 0°, 45°, 90°, and 135°. Further, various statistical descriptors can be extracted from the co-occurrence matrices as the texture representation of images. Haralick et al. Suggested 14 descriptors, including the angular second moment, contrast (variance, difference moment), correlation, and others. Each descriptor represents one texture property. For example, the angular second moment shows proximity of the co-occurrence matrix distribution to the diagonal matrix distribution, which corresponds to the texture homogeneity. The correlation measures the number of local variations of the intensity level, i.e., "diversity of colors" of the image. However, the use of all 14 descriptors, together with the necessity of constructing several co-occurrence matrices for one image (for different offset vectors), would make any image retrieval (or classification) system extremely slow in view of the great amount of calculation needed to process each image. Therefore, many works are devoted to selecting those statistical descriptors derived from the co-occurrence matrices that describe texture in the best way. The following descriptors are considered to be the most discriminative ones and are used more frequently than others:

1. Mean = $mu\_i = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} i * P[i][j]$

2. Variance = $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - mu\_i) \string^2 * P[i][j]$

3. Skewness = $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ((P[i][j]-mu\_i) / (sd\_i))\string^3$

4. Entropy = $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P[i][j] * \log(P[i][j])$

5. Angular Second Moment= $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (P[i][j])\string^2$

6. Contrast = $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i-j)\string^2 * P[i][j]$

7. Homogeneity = $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 1 / (1+ (i-j)\string^2) * P[i][j]$

8. Standard Deviation = (variance)^1/2

9. Correlation = $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ((i-mu\_i) * (j-mu\_j) * P[i][j]) /(sd*sd)$

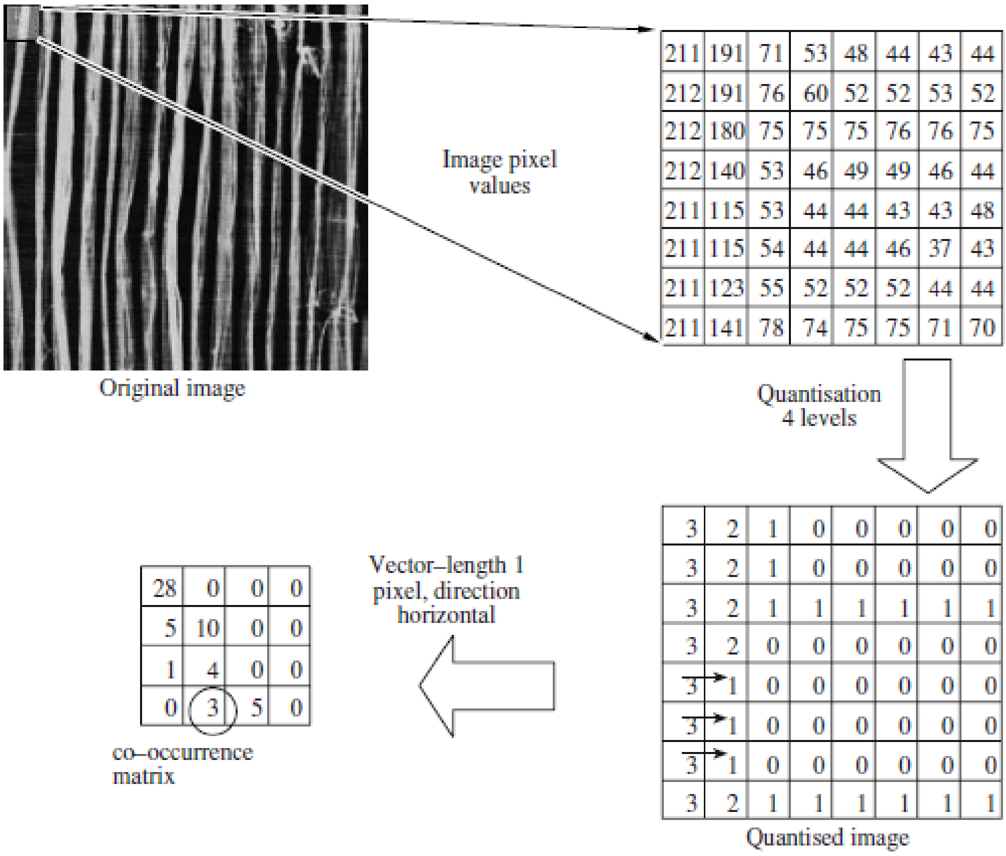


Figure 2.24: An example calculation of the co-occurrence matrix for a region of the image with (Δx, Δy) = (1, 0), [17]

2.6.1.2 Gabor Filter

The *Gabor filter* has been widely used to extract image features, especially texture features. It is optimal in terms of minimizing the joint uncertainty in space and frequency, and is often used as an orientation and scale tuneable edge and line (bar) detector. There have been many approaches proposed to characterize textures of images based on Gabor filters [19]. The basic idea of using Gabor filters to extract texture features is as follows.

A two dimensional Gabor function $g(x, y)$ is defined as:

$$G_{\theta,f}(x,y) = \exp\left(-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right) \cdot \cos(2\pi f x_\theta)$$

with,

$$\begin{cases} x_\theta = x \sin\theta + y \cos\theta \\ y_\theta = x \cos\theta - y \sin\theta \end{cases},$$

where, f gives the frequency of the sinusoidal plane wave at an angle θ with the x - axis and σx and σy are the standard deviations of the Gaussian envelope along the x and y axes, respectively.

Basically, Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency and a specific direction [2][27]. Filter $G_{\theta, f}(x, y)$, computed in equations above; optimally captures both local orientation and frequency information from a digital image. Each image is filtered using the Gabor function, at various orientations, frequencies and standard deviations.

2.7 Shape

Shape features of objects or regions have been used in many content-based image retrieval systems. Construction of the feature vector representing shape of an object on the image can be divided into several stages. The first stage is image segmentation; the second stage is shape representation in terms of abstractions; then, description of the result obtained in terms of a finite set of number parameters composing the feature vector (shape parameterization). On each stage, different algorithms are used. Note that the method of shape parameterization depends obviously on the method of shape representation, whereas the segmentation and extraction of objects is a separate task. The method of image segmentation consists in determining homogeneous regions on the image (or heterogeneities, as boundaries between the regions). The regions obtained are described by means of the shape feature vectors. Further in this section. The simplest shape features are centre of gravity, area, direction of the principal axis, perimeter, and eccentricity. These parameters are easy to determine, but they have limited capabilities. For more accurate search by shape, more complicated methods are used, which make it possible to describe a Figure in more detail. Methods for representing and describing shapes can be divided into two groups: external methods, which represent the region in term of its external characteristics (its boundary), and internal ones, which represent the region in terms of its internal characteristics (the pixels comprising the region). The former are also referred to as boundary-based, since they use only the outer boundary of the shape, and the latter are called region-based, since they use information on the entire shape region.

2.7.1 Shape Based Feature Extraction Techniques

The shape features can be extracted by various boundary based (edge detection based) or by the region based techniques. These features are generally invariant to the transformations; hence provide desirable patterns to be recognised even in case of image transformations.

2.7.1.1 Moment Invariants

Classical shape representation uses a set of *moment invariants*. If the object *R* is represented as a binary image, then the central moments of order *p+q* for the shape of object *R* are defined as:

$$\mu_{p,q} = \sum_{(x,y)\in R} (x - x_c)^p (y - y_c)^q$$

where (*xc*, *yc*) is the centre of object. This central moment can be normalized to be scale invariant

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}, \quad \gamma = \frac{p+q+2}{2}$$

Based on these moments, a set of moment invariants to translation, rotation, and scale can be derived[17]

$$\phi_1 = \mu_{2,0} + \mu_{0,2}$$

$$\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2$$

$$\phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2$$

$$\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2$$

$$\phi_5 = (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})\left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2\right] + (\mu_{0,3} - 3\mu_{2,1})(\mu_{0,3} + \mu_{2,1})\left[(\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{3,0} + \mu_{1,2})^2\right]$$

$$\phi_6 = (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2] + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1})$$

$$\phi_7 = (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2})\left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2\right]$$

2.7.1.2 Grid Based

Grid Based method is an easy to compute method used mainly for the description of object shape. The basic idea of the method is as follows: (1) a grid with cells of certain size is superimposed on the object, and (2) the cells of the grid are traversed from the right to the left and from top to bottom; if the cell covers at least a part of the object, it is associated with number 1; otherwise, with 0 [3]. The sequence of zeros and ones obtained in this way describes the shape of the object. Let us illustrate this by an example [17]. The objects depicted in Figure 2.25 are associated with the following vectors:

(a) 001111000011111111111111111111111 111110111 0111000011
(b) 001100000011100000111100000111101111 111111110 001111000.

The authors of the method proposed to determine similarity of two figures in terms of the number of cells covered by only one figure. In other words, the measure of similarity in this feature space is defined as a sum of elements of the vector that is the result of the XOR operations on the two original vectors. It should be noted that this shape representation is not invariant with respect to rotation and scaling. Invariance of the representation is achieved through preliminary normalization of the Figure relative to its major axis. Placing the figure in such a way that its major axis is parallel to the *X* axis (Figure. 2.26), we get invariance with respect to rotation. Before comparing two figures, we change their scales such that their major axes have equal length. This will make the representation scale invariant.

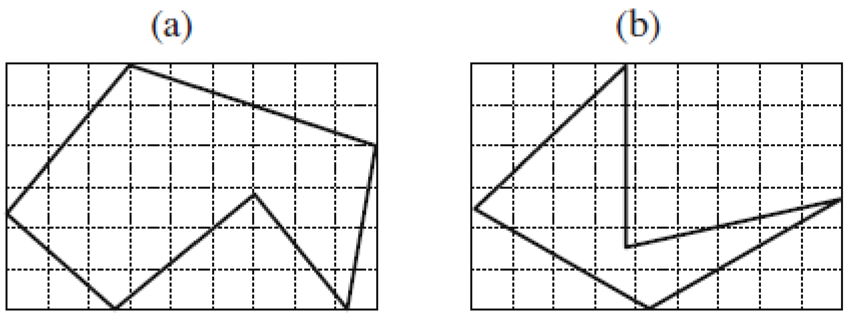


Figure 2.25: Imposing Figures on the grid to get index

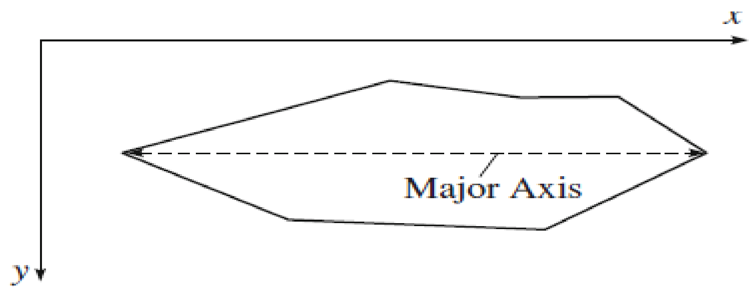


Figure 2.26: Normalisation of a Figure with respect to the major axis

CHAPTER 3

PROPOSED SYSTEM: PRISM

- ***ANALYSIS & DESIGN***

CHAPTER 3
PRISM - ANALYSIS & DESIGN

The various image processing techniques for feature extraction have been applied in this project. The proposed system performs the task of pattern recognition and image analysis (image understanding) using feature extraction techniques. It has hence been termed as *PRISM* – *P*attern *R*ecognition & *I*mage *A*nalysis *M*achine.

3.1 Project Definition

Image Understanding is becoming an important requirement of many critical application of special importance to various defence and medical applications, for example. The proposed system suggests the application of the feature extraction techniques to the image understanding and content based image retrieval.

Image understanding is used to analyse an input image and generate an inference regarding the scene represented in the image. In *PRISM* the image understanding module is used in defence application. It is used to locate the air bases constructed by various militant groups settled around the high security areas. It distinguishes a government established airport from the one built be militant groups for air attacks. It is also used to distinguish an airport hangar from a runway. The similar approach can also be used in case of shipyards by providing a different image database and training the system to generate the decision accordingly. The image understanding scheme proposed here uses object recognition i.e. image retrieval to identify the aircrafts. The retrieved results are used to generate the inference with the help of the decision trees.

Content Based Image Retrieval is used to analyse the input image and retrieve similar results from the database based on the content of the image, which is fetched through the feature extraction techniques. The Image retrieval process is also used to identify objects within an image, a concept known as object recognition. This object recognition is used as a major building block for the image understanding system.

3.2 System Requirement Analysis

The various requirements of the system are identified in this section. It includes the identification of requirements, the preliminary investigation, the hardware and the software requirements of the system. It also specifies the external interfaces to the system and describes the functional requirements of the system.

3.2.1 Identification of requirements

Early techniques were not generally based on visual feature but on the textual annotation of image. In other words, images were first annotated with text and then search using a text based approach from traditional database management systems. Since automatically generating descriptive texts for a wide spectrum of images is not feasible, most text based image retrieval systems require manual annotation of images. As a result, it is difficult for the traditional text based methods to support a variety of task dependent queries.

Also, the earlier image understanding techniques have been proposed to identify objects, rather than combining the results of various objects to analyse the entire scene. The earlier used techniques make it difficult to combine different characteristic features of the image and their in present objects.

3.2.2 Preliminary Investigation

The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the system in all respects.

Due to the enormous increase in image database sizes, as well as its vast deployment in various applications, the need for such a pattern recognition system arose.

3.2.3 Feasibility Study

Technical Feasibility

This is concerned with specifying equipment and software that will successfully satisfy the proposed image understanding and retrieval system using feature extraction techniques. It has been developed on Java technology and tools which are easily available and open source.

Economic Feasibility

In many software application areas it is often more cost effective to acquire than develop computer software. The System is economically feasible as open source tools are used.

Operational Feasibility

It is mainly related to human and organizational aspects.

PRISM can be used in the organization in which Image Retrieval is the main and important task. It would improve the overall working of the Image retrieval. They can handle the images in an effective manner.

The project can also be used in defence applications. The system can be used to identify illegal settlements (airbase and ship yard) near the borders by the terrorist groups. It would help identify such locations after getting input feed through the satellite images. Necessary steps can then be taken up by the concerned authorities.

3.2.4 External Interfaces

- *PRISM* helps in the feature extraction on the basis of color, shape and texture of an image and using the features for better understanding of the image.
- User will provide input in the form of an image.
- The result will be generated in time in accordance with the size of the image.
- Error messages are generated in case format other than ‘bmp’ is chosen.

3.2.5 Hardware Requirements

S.No	Processor	RAM	Disk Space
1	Any Pentium Version >4	2 GB	2 GB for PC
2	Pentium Version>4 for Postgre SQL	2 GB	2 GB for PC

Table 3.1: Hardware Requirements of *PRISM*

3.2.6 Software Requirements

S.No	Description	Architect	Operating System
1	Database	Postgre SQL	Windows XP and above
2	User Interface	Java Swing	Windows XP and above

Table 3.2: Software Requirements of *PRISM*

3.2.7 Memory Constraint

The system needs 2GB RAM or above.

3.2.8 Functional Requirements

- The system uses a graphical user interface for easy and intuitive way of getting input in the form of images from the user.
- The system uses visual features such as color, texture and shape of images for feature based extraction.
- It helps in understanding the input image and identifying those of the illegal settlements.
- It helps in fine definition of the image as of a storage unit of aircrafts or that of an area used for flying aircrafts.
- There is a database used for storing images and the results for the query images.
- The system uses pattern recognition and image processing mechanism for its applications.
- The system uses decision tree approach for understanding the contents of the image.
- The product has separate modules for database such as create, import and delete.

3.2.9 Use Case Diagram

The use case diagrams are used to represent the functionalities of the system with respect to the different actors of the system. The major actors of the system are the system as a whole and the external user of the system.

Following sub section describes the functionalities of the system with respect to the actors.

Use Case Template

The use cases in the system are described in the following use case templates.

Use Case: Create Database

1.	**Brief Description:** Creates the database for pattern recognition
2.	**Actors:** System, Database
3.	**Flow of Events** **3.1 Basic Flow:** Database which includes the tables and folders is created. **3.2 Alternate Flow:** In case connection to database is not established, the database is not created and the system is disabled.
4.	**Special Requirements:** The Postgre SQL must be installed on the system. Also the name of the default database and the username and password of the Postgre SQL should be known.
5.	**Pre- conditions:** The database with the same name should not be pre-existing
6.	**Post- conditions:** The tables and the folders required in the database are created
7.	**Extension Points:** Connect To Database

Use Case: Connect To Database

1.	**Brief Description:** Establishes the connection with the database
2.	**Actors:** System, Database
3.	**Flow of Events** **3.1 Basic Flow:** Authentication of username, password checked; Connection to database established; SQL statement initialized. **3.2 Alternate Flow:** In case connection to database is not established, the system, PRISM, is disabled.
4.	**Special Requirements:** The Postgre SQL must be installed on the system. Also the username and password of the Postgre SQL should be known.
5.	**Pre- conditions:** The database with the same name should not be pre-existing
6.	**Post- conditions:** The tables and the folders required in the database are created
7.	**Extension Points**

Use Case: Read Image

1.	**Brief Description:** Reads the image
2.	**Actors:** System
3.	**Flow of Events** **3.1 Basic Flow:** Reads the image selected by the system **3.2 Alternate Flow:** In case image is not read, the image data matrix contains null value and the result image is initialised with a null value.
4.	**Special Requirements:** The system must have the Java JAI API library included in the project.
5.	**Pre- conditions:** Some image should be selected by the system
6.	**Post- conditions:** The Image data matrix contains the RGB values at the various pixels and the result image is initialised with the image.
7.	**Extension Points**

Use Case: Feature Extraction

1.	**Brief Description:** The feature parameters of the image are extracted
2.	**Actors:** System, Database
3.	**Flow of Events** **3.1 Basic Flow:** Feature parameters of the image are calculated and extracted from the database, according to the specified technique. **3.2 Alternate Flow:** In case feature parameters are not calculated, then error message is generated.
4.	**Special Requirements:** The Postgre SQL must be installed on the system for storing the feature parameters.
5.	**Pre- conditions:** The technique and image must be selected for which the features are to be extracted.
6.	**Post- conditions:** Feature parameters of the image are calculated and extracted.
7.	**Extension Points**

Use Case: Understand Image

1.	**Brief Description:** Input image is analysed for various application considered for evaluation.
2.	**Actors:** System, Database
3.	**Flow of Events** **3.1 Basic Flow:** Various objects are recognised in the input image using feature extraction. Then the shortest relative distance is calculated between the various recognised objects. **3.2 Alternate Flow:**
4.	**Special Requirements:** The Postgre SQL must be installed on the system for object recognition.
5.	**Pre- conditions:** Image understanding application must be selected.
6.	**Post- conditions:** The input image is analysed.
7.	**Extension Points**

Use Case: Decision Generation

1.	**Brief Description:** generates decision about the image using decision tree
2.	**Actors:** System
3.	**Flow of Events** **3.1 Basic Flow:** Estimation values are assigned to the system according to the objects recognised and the relationship between them. The decision is generated about the image using decision tree approach. **3.2 Alternate Flow:** In case the image is not clearly recognised, then a low estimation value is assigned and the image analysis generates an error message.
4.	**Special Requirements:** The objects are clearly identified.
5.	**Pre- conditions:** The estimation values should be known before decision tree is to be to be built.
6.	**Post- conditions:** A conclusive decision is generated about the image.
7.	**Extension Points**

Use Case: Computing Result

1.	**Brief Description:** The final result is computed after the various applications of PRISM
2.	**Actors:** System, Database
3.	**Flow of Events** **3.1 Basic Flow:** The result data matrix obtained is converted to the result image and displayed to the user for analysis. **3.2 Alternate Flow:** In case connection result data matrix is not computed, then the result is computed as null.
4.	**Special Requirements:**
5.	**Pre- conditions:** The processing of the PRISM application should have computed successfully to yield data matrix.
6.	**Post- conditions:** The result image is displayed as the result.
7.	**Extension Points**

Use Case: Import Images

1.	**Brief Description:** Imports images for the pattern recognition application
2.	**Actors:** User , Database
3.	**Flow of Events** **3.1 Basic Flow:** tables required for the image selected are created at the module and the image is saved in the database folder. **3.2 Alternate Flow:** In case connection to database is not established, the images are not imported.
4.	**Special Requirements:** The Postgre SQL must be installed on the system
5.	**Pre- conditions:** The database with the same name should not be pre-existing
6.	**Post- conditions:** The tables and the folders required in the database corresponding to the selected image are created
7.	**Extension Points**

Use Case: Object Based Recognition

1.	**Brief Description:** Performs the object based recognition
2.	**Actors:** User, Database
3.	**Flow of Events** **3.1 Basic Flow:** The input image, query object and technique selected are used to fetch the object parameter and recognize the objects in the input image. **3.2 Alternate Flow:** In case object based recognition is not performed, the result image is presented as a null image.
4.	**Special Requirements:** Object based recognition application is selected by the user
5.	**Pre- conditions:** The input image, query object and technique used must be selected
6.	**Post- conditions:** Objects similar to the query objects are retrieved from the image.
7.	**Extension Points**

Use Case: Target Based Recognition

1.	**Brief Description:** Performs the target based recognition
2.	**Actors:** User, Database
3.	**Flow of Events** **3.1 Basic Flow:** The input image and technique selected are used to fetch the target parameter and recognize the similar images are retrieved from the image database **3.2 Alternate Flow:** In case target based recognition is not performed, the result image is presented as a null image.
4.	**Special Requirements:** Target based recognition application is selected by the user. Also the Postgre SQL must be installed on the system.
5.	**Pre- conditions:** The input image and technique used must be selected
6.	**Post- conditions:** Images similar to the input image are retrieved from the database
7.	**Extension Points**

Use Case: Image Understanding

1.	**Brief Description:** Understands the input image
2.	**Actors:** User, Database
3.	**Flow of Events** **3.1 Basic Flow:** Understand Image module is called and the decision is generated by the system. **3.2 Alternate Flow**
4.	**Special Requirements:** The Postgre SQL must be installed on the system
5.	**Pre- conditions:** Image Understanding application must be selected
6.	**Post- conditions:** The input image is analysed.
7.	**Extension Points**

Use Case: Delete Images

1.	**Brief Description:** Deletes the images in the database
2.	**Actors:** User, Database
3.	**Flow of Events** **3.1 Basic Flow:** The image for the particular type of application to be deleted is available, the tables with the image name are correspondingly deleted. **3.2 Alternate Flow:** In case connection to database is not established, the database is not created and the system is disabled.
4.	**Special Requirements:** The Postgre SQL must be installed on the system. Also the database with the same name should not be pre-existing
5.	**Pre- conditions:** The application for which the image is to be deleted is available.
6.	**Post- conditions:** The tables and the images in the database according to the one image selected are deleted.
7.	**Extension Points**

Use Case: Save Result

1.	**Brief Description:** Saves the result image
2.	**Actors:** User, Database
3.	**Flow of Events** **3.1 Basic Flow:** The result data matrix is written to a .bmp image. **3.2 Alternate Flow:** In case the header of the image file is not clearly written, then the result image is not saved.
4.	**Special Requirements:** The Postgre SQL must be installed on the system
5.	**Pre- conditions:** Result image data matrix is available
6.	**Post- conditions:** The result image is saved in .bmp format.
7.	**Extension Points**

Use Case: Exit

1.	**Brief Description:** Exits from the system
2.	**Actors:** User
3.	**Flow of Events** **3.1 Basic Flow:** The flow terminates from the system and the system is exited. **3.2 Alternate Flow**
4.	**Special Requirements**
5.	**Pre- conditions:** The user has called the exit module or some fault has occurred in the system.
6.	**Post- conditions:** The system is exited
7.	**Extension Points**

The Use Case Diagrams for describing the system are presented in Figure 3.1 and Figure 3.2.

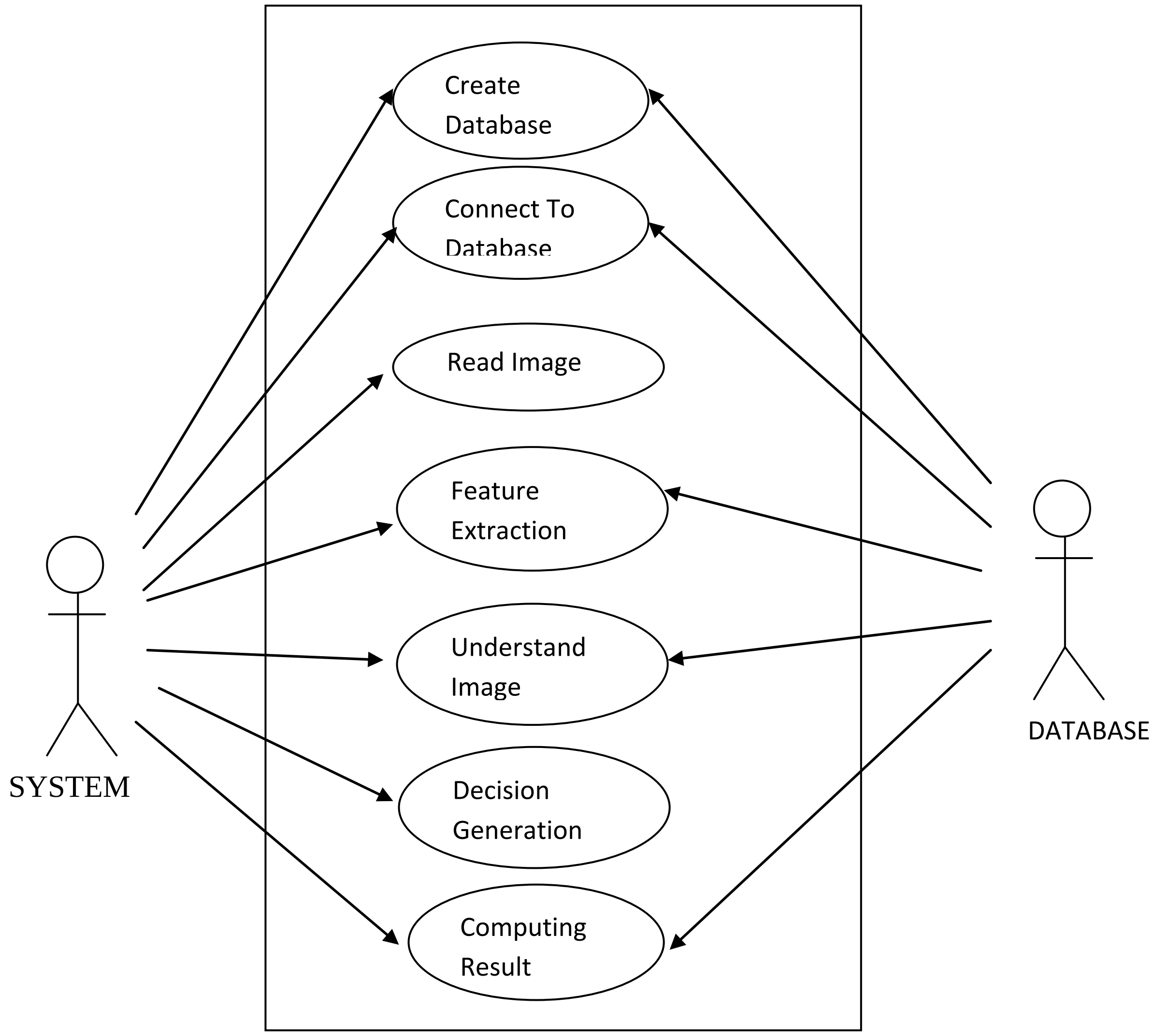


Figure 3.1: Use case diagram: system

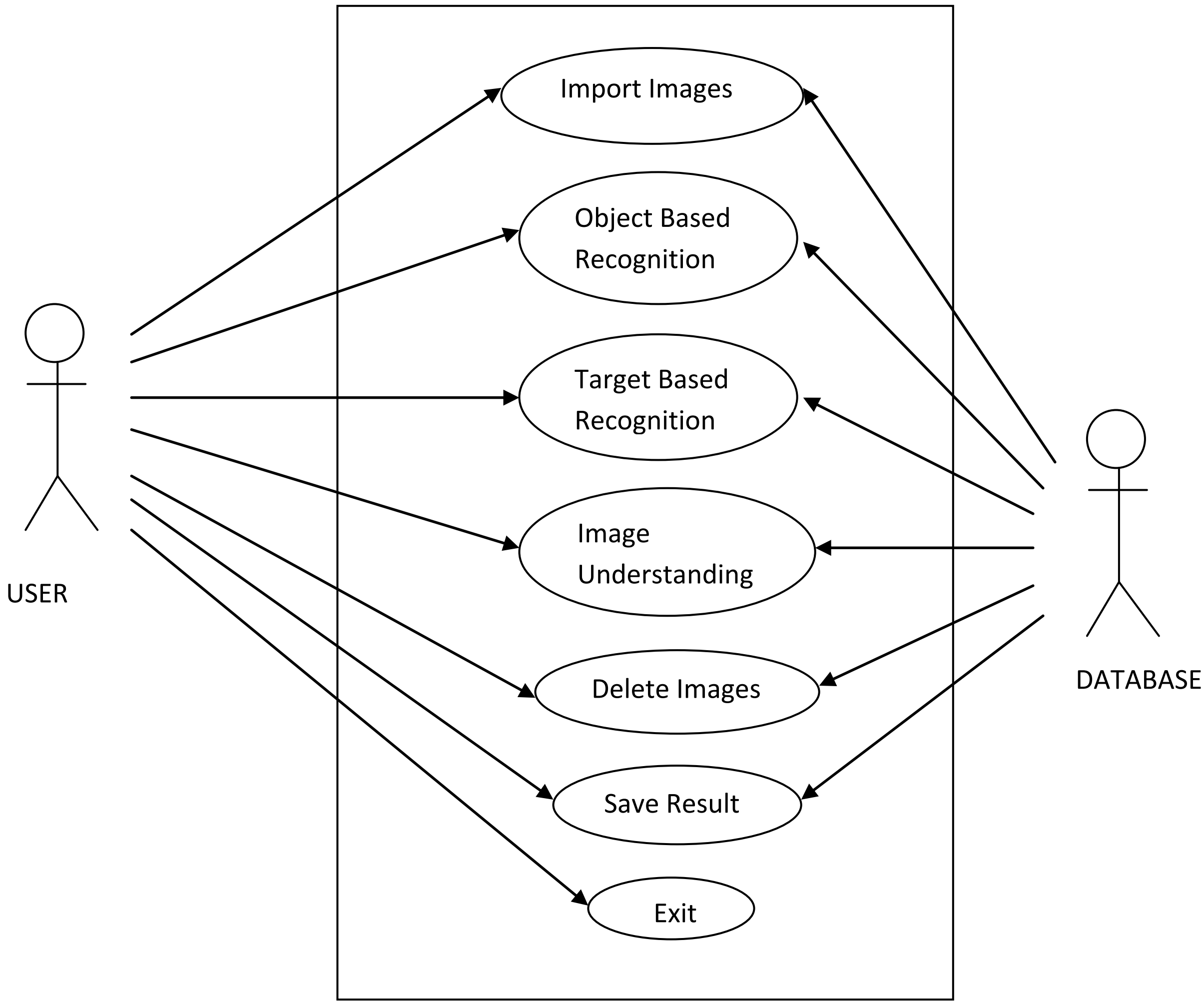


Figure.3.2: Use case diagram: user actor

3.2.10 Performance Requirements

- The system should support only one terminal.
- The original image should be of bmp format and size of the image is fixed only for some applications.
- Result should be generated in the best time possible

3.2.10.1User Interface

- The system should have a GUI which should use easy and intuitive ways of getting input from the user.
- The user should have the option of saving the result obtained.

3.2.11 Logical Database Requirements

- A Database should be maintained to store images for all applications.
- There should be a functionality to import and delete the images from database.

3.3 System Design

This section deals with the design phase of the system. It deals with the various data flow diagrams, activity diagrams which represent the various design details of the system. The layered architecture of the system is presented here. The various functional components of the system are also identified in the design analysis.

3.3.1 Data Flow Diagrams

The data flow diagrams represent the system as integrated modules. The detailed view of each module as the higher order data flow diagrams are represented. The system level DFD describes the input-output for the system as a whole. Further level of DFD represent the detailed processes and their data flow.

To clarify the system requirements and identifying major transformations the bubble chart is presented. This will act as the starting point of the design phase. It functionally decomposes the requirements specifications down to the lowest level of details.

Level-0- DFD

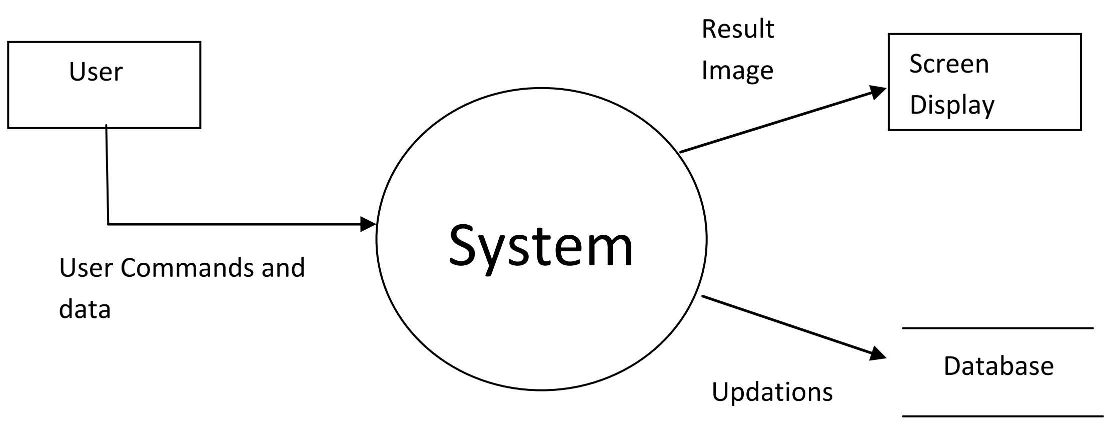


Figure 3.3: Level -0- DFD

Level -1- DFD

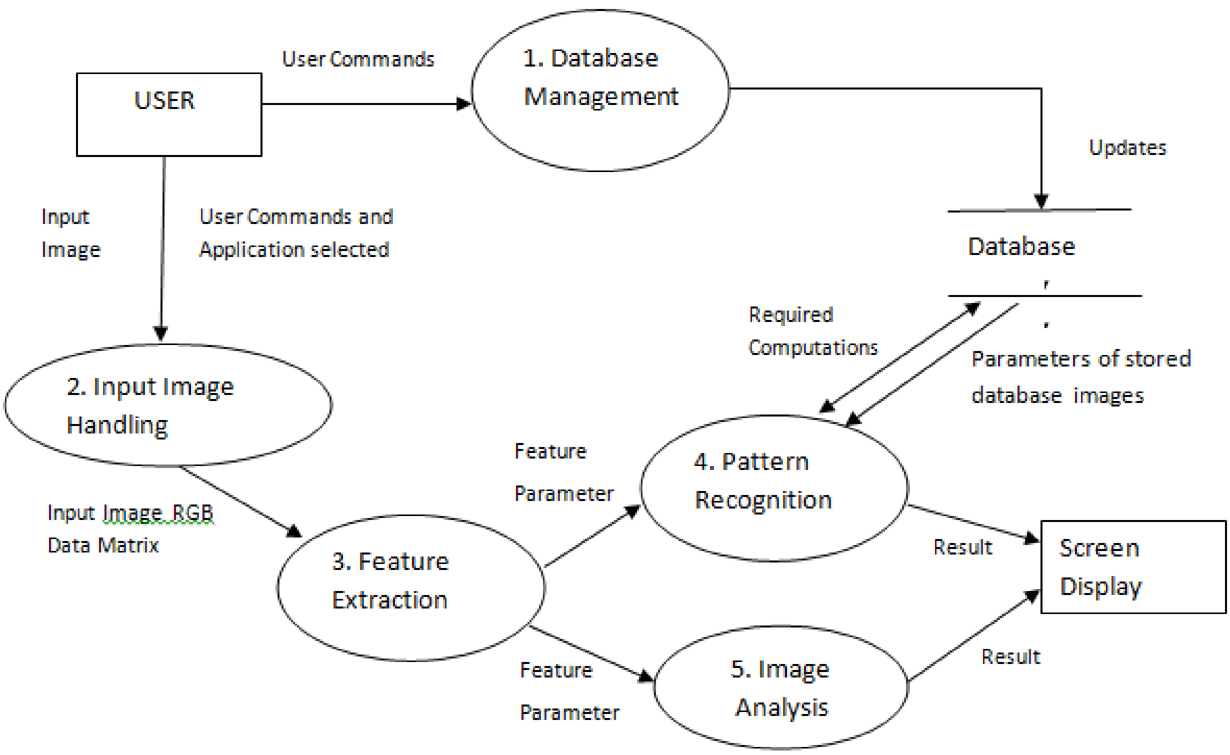


Figure 3.4: Level-1-DFD: System Description

Level-2-DFD

Process 1. Database Management

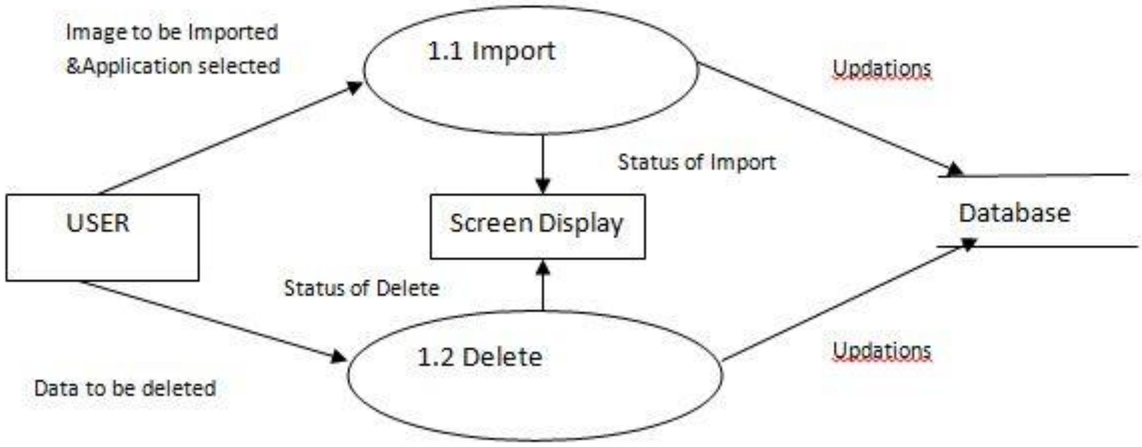


Figure 3.5: Level 2 DFD: Database Management

Process 2. Image Handling

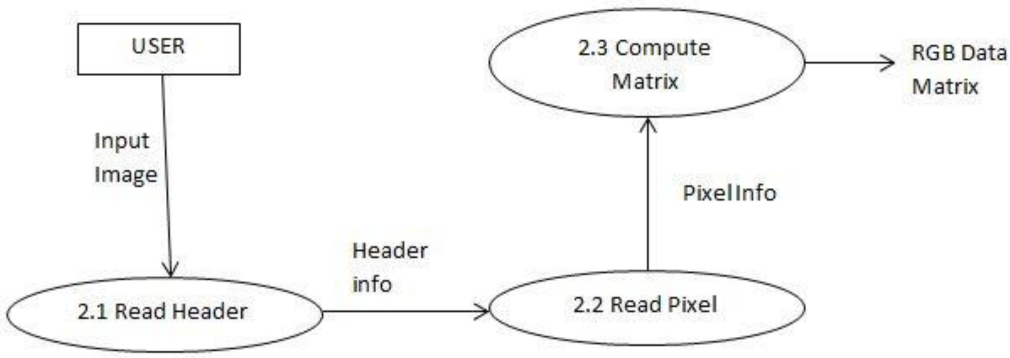


Figure 3.6: Level 2 DFD: Image Handling

Process 3. Feature Extraction

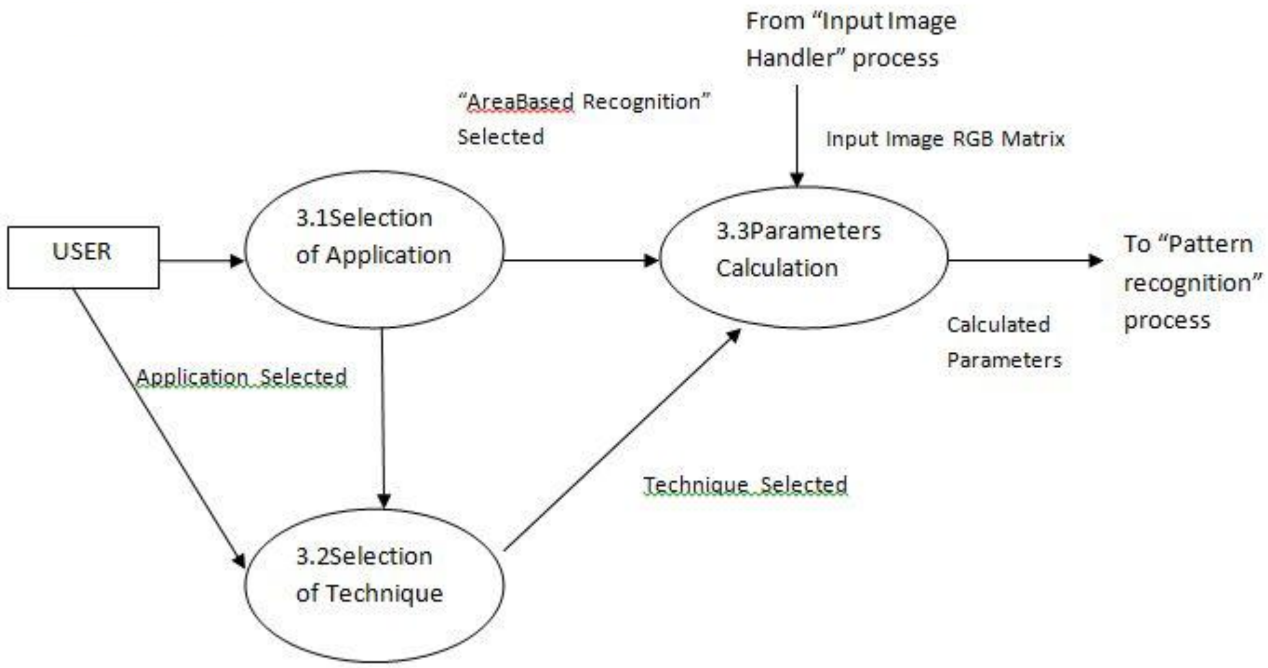


Figure 3.7: Level 2 DFD: Feature Extraction

Process 4. Pattern Recognition

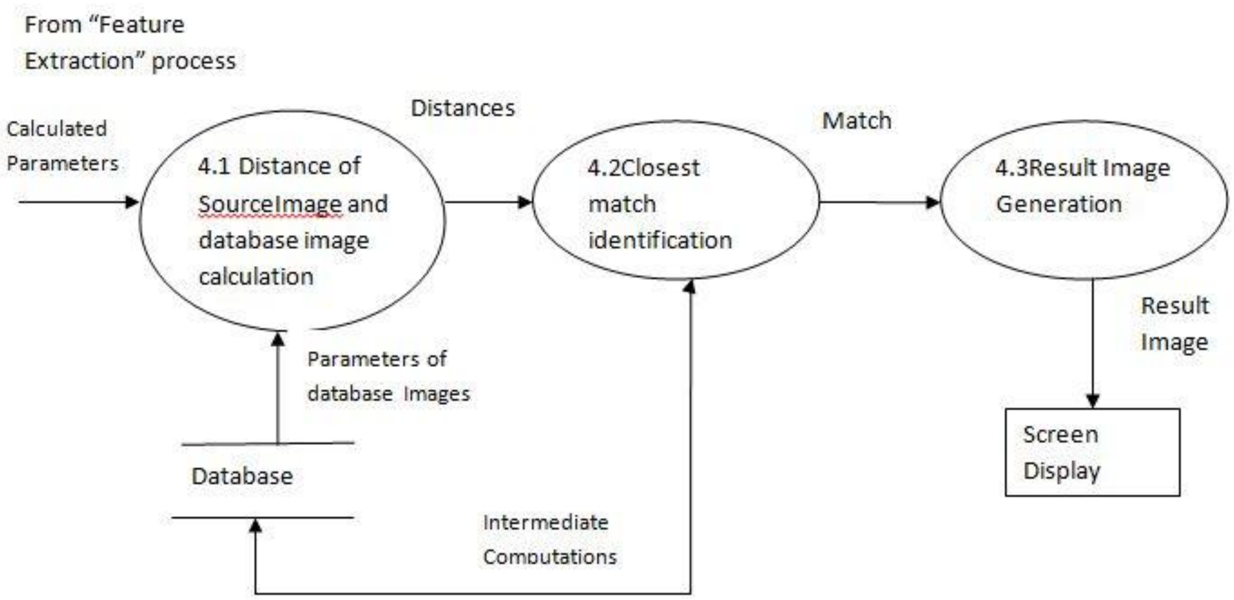


Figure 3.8: Level 2 DFD: Pattern Recognition

Process 5. Image Analysis

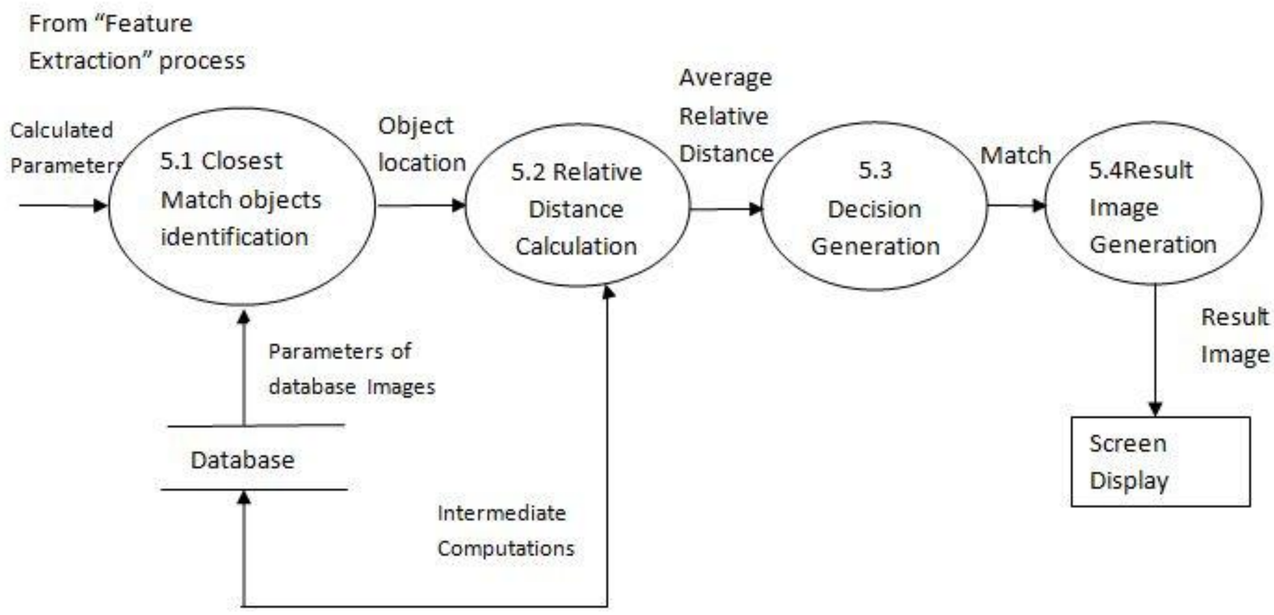


Figure 3.9: Level 2 DFD: Image Analysis

3.3.2 Activity Diagram

Object based recognition

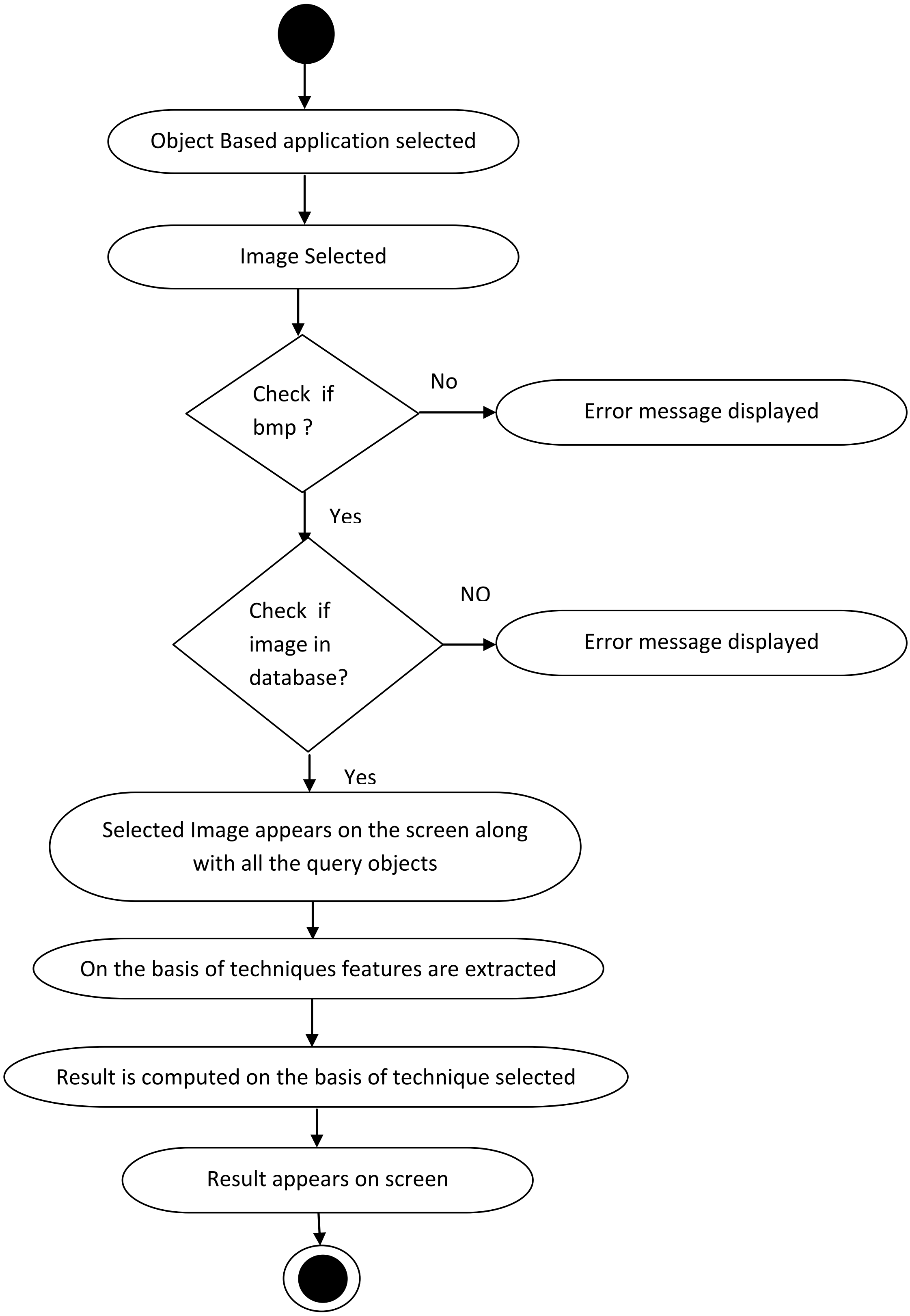


Figure 3.10: Object Based Recognition Activity Diagram

Target based recognition

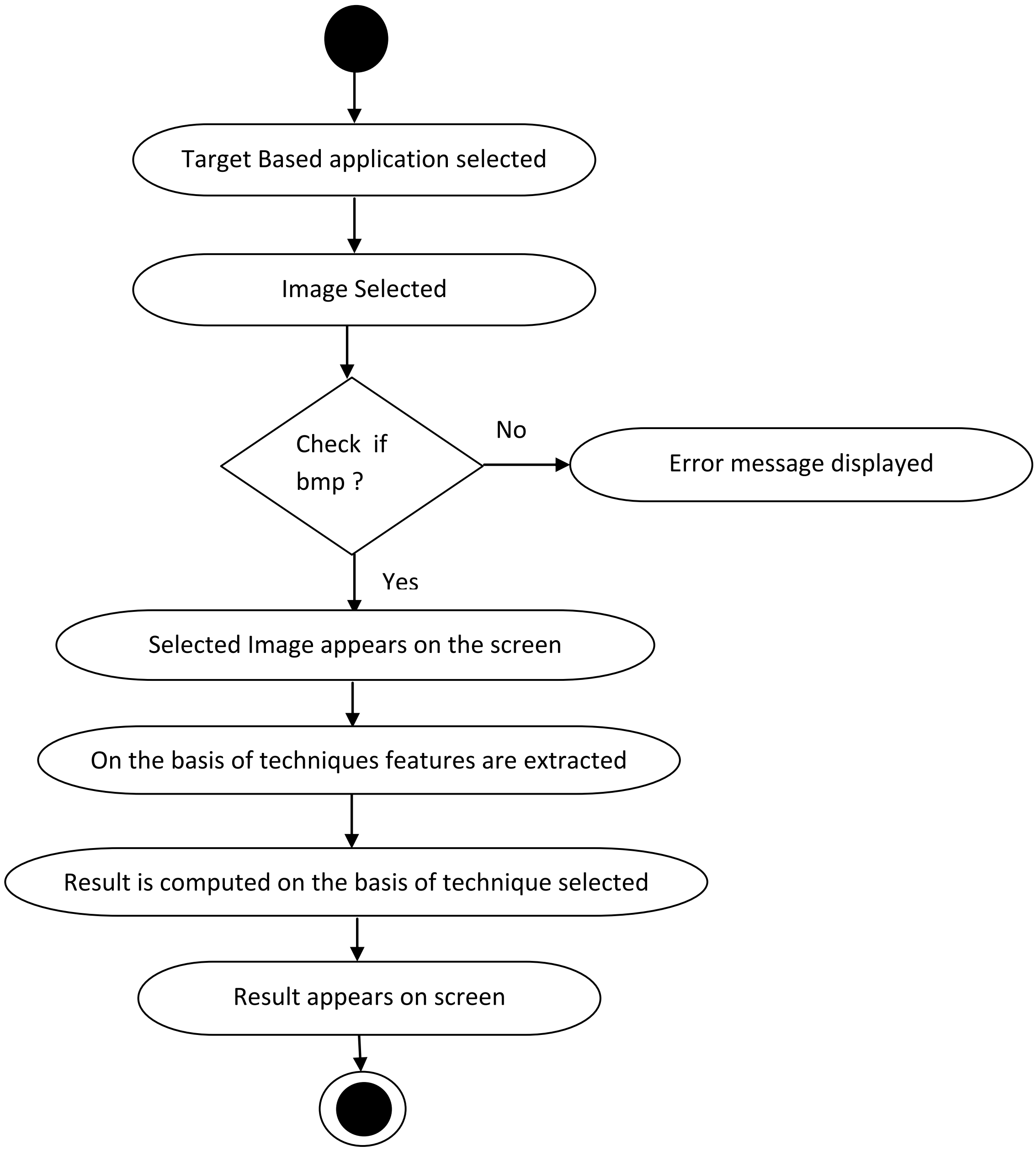


Figure 3.11: Target Based Recognition Activity Diagram

Image Understanding

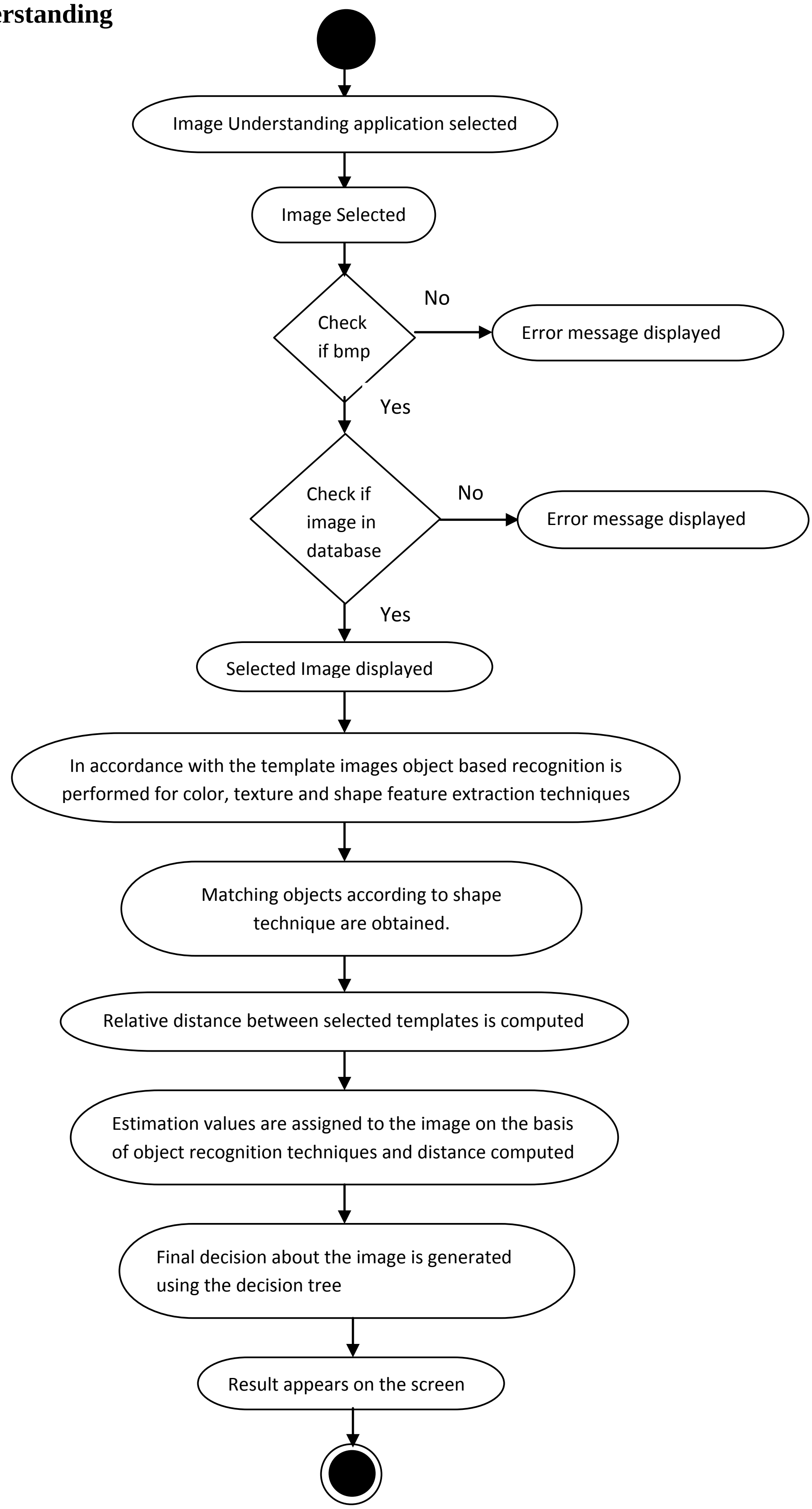


Figure 3.12: Image Understanding Activity Diagram

Connection to Database

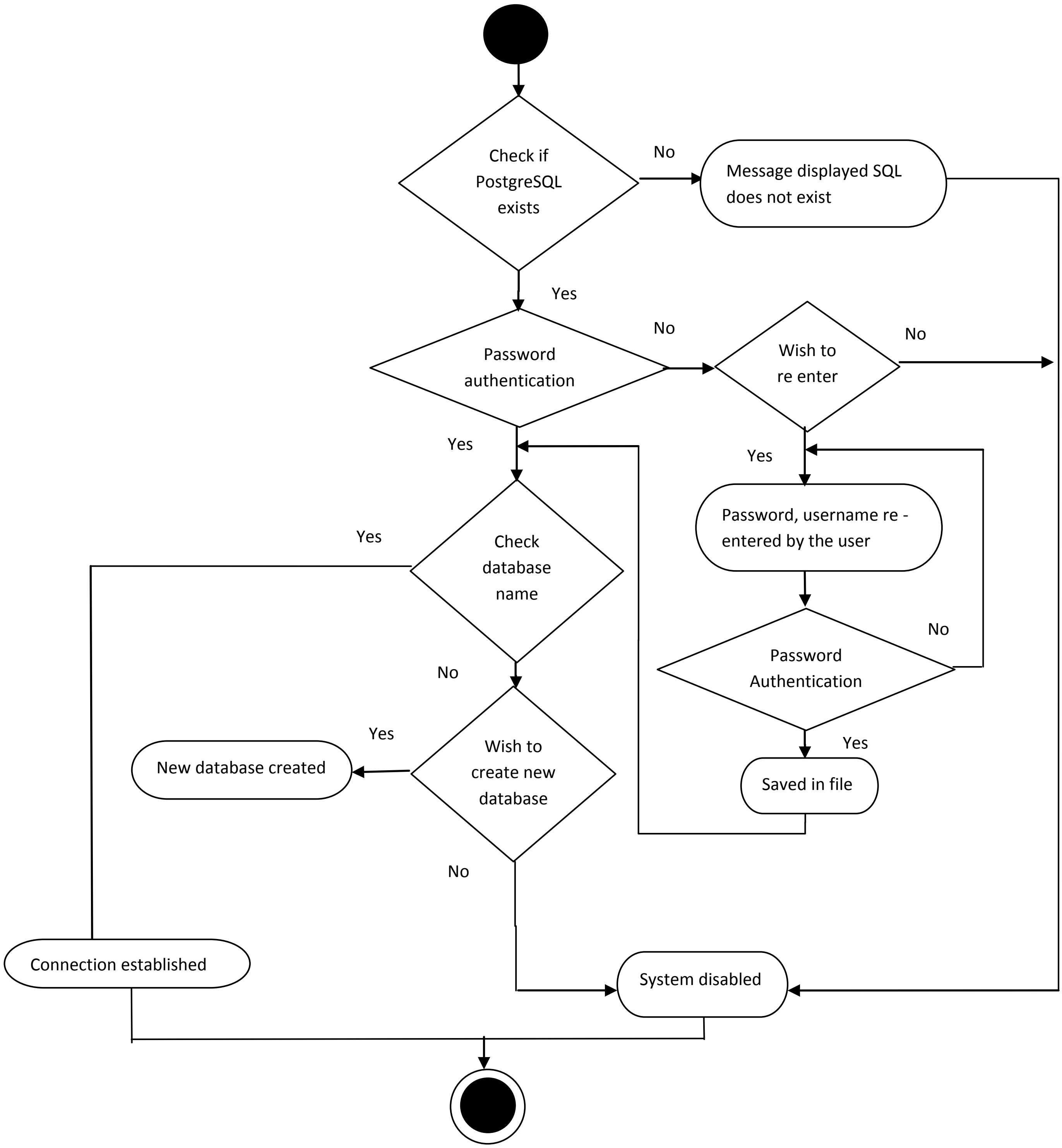


Figure 3.13: Connection to Database Activity Diagram

3.3.3 Layered Architecture & Block Diagram of System

The software developed lies in category of two tier application.

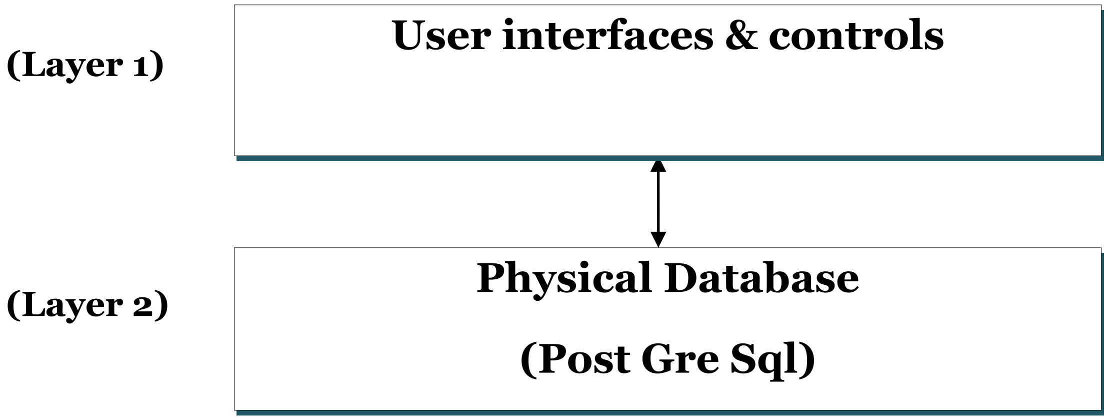


Figure 3.14: Layered Architecture

The block diagram of the system is as represented in Figure 3.15.

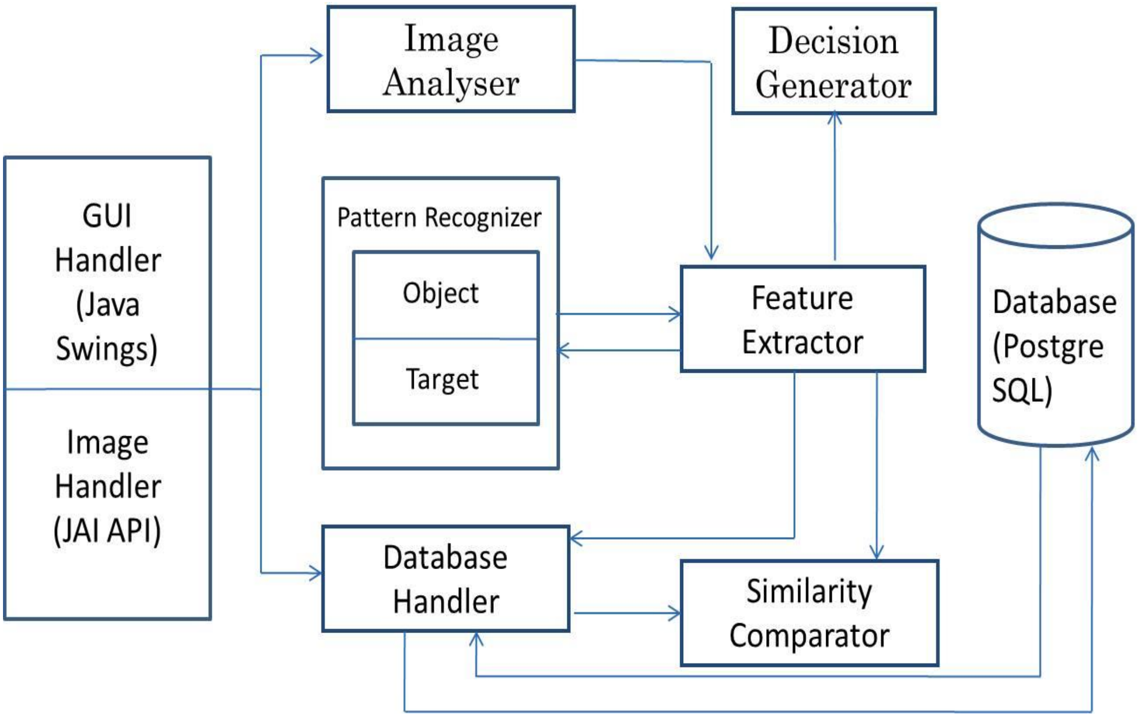


Figure 3.15: PRISM Block Diagram

The block diagram describes the major components of *PRISM*.

Image Handler

This module concerns with converting the image to its RGB Data matrix. It also concerns with saving the result image data matrix in the form of image. This module is a major component for the other modules of the project.

Feature Extraction

This module concerns with extracting the parameters of a given image on the basis of certain features. The project is using the following features

- **Color**

  `Technique used: Color Moments

  Fuzzy Color Moments

  Color Histogram

- **Texture**

  Technique used: Grey Level Co-Occurrence Matrix

  Gabor Filter

- **Shape**

  Technique used: Moment Invariant

Pattern Recognition

Pattern recognition is the scientific discipline that concerns the description and classification of the patterns. It aims to recognize patterns based on their features. It mainly comprises of the following modules:

- Object based pattern recognition: A given query object is searched in the given image and the most resembling part of the image is highlighted. This matching is done on the basis of the parameters extracted in the module "Feature Extraction".
- Target Based Recognition: A given image is searched in the database on the basis of the parameters extracted in the module "Feature Extraction". The closest match of the given image and among the set of images in the database appears as result on the screen.

Image Analysis

The image analysis is the image processing domain which analyses the image and provides required information about the image. In this project this module provides information about the scene represented in the image. It is used to analyse the relative distance between the objects identified in the image. It also analyses the ground texture (metalled road, sandy plain, etc.) represented in the image. Using the information about the objects, the texture and other information gathered from the image, the decision tree approach is used with expectations values assigned to the image according to the various characteristic features of the image. The image analysis (or image understanding approach) is highly preferable to understand the image and use it for suitable critical applications.

Database Management

This module deals with the updation and fetching the data from the database. Updation includes the following:

Import Images: This involves importing the images to the database

Delete Images: This involves deleting the images from the database

Images comprises following types:

Following used in Object Based Recognition

Database FE Image: The image on which the search is done. It can be of any size.

Query Object Image: The object which is to be searched whose size should be of 50X50

Following used in Target Based Recognition

Database Target Image: The image which is in the database used for target based recognition. Its size is 400 x 400.

3.4 Database Description

Folders:

- Query Object : stores all the query objects
- Database FE Image : stores all Database FE Images
- Database Target Image: stores all Database Target Image

Tables:

- Table Name: *queryobjectname_table*
  Attributes: queryobjectname character varying (50)
  Table Description: stores names of all query objects

- Table Name: *databsefeImagename_table*
  Attributes: databasefeimagename character varying (50) NOT NULL, PRIMARY KEY
  Table Description: contains the names of all Database FE Images

- Table Name: *databasetargetimagename_table*
  Attributes: databasetargetimagename character varying (50) NOT NULL, PRIMARY KEY
  Table Description: contains the names of all the database Target Image names

- Table Name: *targetimagecolorhistogram*
  Attributes: targetimagename character varying (50) NOT NULL, PRIMARY KEY,
  r_bin1, g_bin1, b_bin1,
  r_bin2,g_bin2, b_bin2,
  .....
  r_bin8,g_bin8,b_bin8 Integer NOT NULL
  Table Description: Contains all the Color Histogram Parameters for all the Database Target Images

- Table Name: *targetimagecolormoments*
  Attributes: targetimagename character varying (50) NOT NULL, PRIMARY KEY,
  r_mean, g_mean, b_mean,
  r_sd,g_sd, b_sd,
  r_skew,g_skew,b_skew Numeric NOT NULL
  Table Description: Contains all the Color Moment Parameters for all the Database Target Images

- Table Name: *targetimagetextureglcm*
  Attributes: databasetargetname character varying (50) NOT NULL, PRIMARY KEY,
  r_contrast, g_contrast, b_contrast,
  r_ homogeniety, g_ homogeniety, b_homogeniety,

r_ dissimilarity, g_ dissimilarity, b_dissimilarity,
r_ energy, g_ energy, b_energy,
r_ correlation, g_ correlation, b_ correlation,
r_ entropy, g_ entropy, b_entropy, Numeric NOT NULL

Table Description: Contains all the Texture GLCM Parameters for all the Database Target Images

- Table Name: *object_template_distance*
  Attributes: i integer, j integer, PRIMARY KEY,
  distance Decimal
  Table Description: Contains the Distance of the templates of the database FE image selected for the object based Application from the query object selected. It is a temporary table dropped at the beginning of the function GetQueryObjectResult before each time the application is re- run.

- Table Name: *databasetarget_source_distance*
  Attributes: targetimagename character varying (50), NOT NULL, PRIMARY KEY,
  distance Decimal
  Table Description: Contains the Distance of the database Target image selected for the target based Application from the other target images in the database. It is a temporary table dropped at the end of the function GetTargetImageResult.

- Table Name: *databasefeimagenamecolorhistogram*
  Eg: Image Name: test.bmp
  TableName: testcolorhistogram
  Attributes: i integer, j integer, NOT NULL, PRIMARY KEY,
  r_bin1, g_bin1, b_bin1,
  r_bin2,g_bin2, b_bin2,
  .....
  r_bin8,g_bin8,b_bin8 Integer NOT NULL
  Table Description: Contains all the Color Histogram Parameters for all the templates for the particular Database FE Image

- Table Name: *databasefeimagenamecolormoments*
  Eg: Image Name: test.bmp
  TableName: testcolormoments
  Attributes: i integer, j integer, NOT NULL, PRIMARY KEY,
  r_mean, g_mean, b_mean,
  r_sd, g_sd, b_sd,
  r_skew, g_skew, b_skew Numeric NOT NULL
  Table Description: Contains all the Color Moment Parameters for all the templates for the particular Database FE Image

- Table Name: *databasefeimagenametextureglcm*
  Eg: Image Name: test.bmp
  TableName: testtextureglcm
  Attributes: i integer, j integer, NOT NULL, PRIMARY KEY,
  r_contrast, g_contrast, b_contrast,
  r_ homogeniety, g_ homogeniety, b_homogeniety,
  r_ dissimilarity, g_ dissimilarity, b_dissimilarity,
  r_ energy, g_ energy, b_energy,
  r_ correlation, g_ correlation, b_ correlation,
  r_ entropy, g_ entropy, b_entropy, Numeric NOT NULL
  Table Description: Contains all the Texture GLCM Parameters for all the templates for the particular Database FE Image

The last few tables are formed for each Database FE Image imported in the database for all the techniques.

3.5 Algorithms Used

The implementation details of these algorithms have been discussed in Appendix 1. The various algorithms used in the project are presented here. First the algorithms for the various feature extraction techniques are represented, followed by the algorithms used in image retrieval and then the ones used by the image understanding modules of the project.

Application Module	**Algorithm Included**
Feature Extraction	Color Moments
	Fuzzy Color Moments
	Color Histogram
	Texture GLCM
	Gabor Filter
	Moment Invariant
Pattern Recognition	Object Based Recognition
	Target Based Recogntion
Image Analysis	Image Understanding
	Relative Distance Measurement
	Decision Tree Algorithm

Table 3.3: Algorithms for various Project Modules

3.5.1 Feature Extraction Techniques

i. Color Moments

1. Image as input.
2. Extract attributes of the image i.e. Height, Width and RGB values.
3. Use extracted RGB values to calculate color parameters i.e.

   mat[i][j][0] for Red

   mat[i][j][1] for Green

   mat[i][j][2] for Blue
4. We calculate three order moments for color parameters.
5. First order moment i.e. means is calculated for R, G and B component of image.

   LOOP i->0 to h-1

   LOOP j->0 to w-1

$$r\_sum = \left(\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} mat[i][j][0] \right)$$

$$g\_sum = \left(\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} mat[i][j][1] \right)$$

$$b\_sum = \left(\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} mat[i][j][2] \right)$$

   here h is the image Height and w is the image Width.

   r_mean=r_sum/(Height*Width)

   g_mean=g_sum(Height*Width)

   b_mean=b_sum/(Height*Width)

6. Second order moment i.e. variance is calculated for R, G and B component of image.

   LOOP i->0 to h

   LOOP j->0 to w

$$R\_SquareSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (mat[i][j][0]-r\_mean)\ \hat{}\ 2$$

$$G\_SquareSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (mat[i][j][1]-g\_mean)\ \hat{}2$$

$$B\_SquareSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (mat[i][j][2]-b\_mean)\ \hat{}2$$

where h is the Height of the image and w is the Width of the image

R_var = R_SquareSumDeviation/(Height*Width);
G_var = G_SquareSumDeviation/(Height*Width);
B_var = B_SquareSumDeviation/(Height*Width);

7. Third order moment i.e. skewness is calculated for R, G and B component of image.

LOOP i->0 to h-1

LOOP j->0 to w-1

$$R\_CubeSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [(mat[i][j][0]-r\_mean)]\ \hat{}3$$

$$G\_CubeSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [(mat[i][j][1]-g\_mean)]\ \hat{}3$$

$$B\_CubeSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [(mat[i][j][2]-b\_mean)]\ \hat{}3$$

Where h is the Height of the image and w is the Width of the image

R_skew=Math.cbrt(R_CubeSumDeviation/(Height*Width));
G_skew=Math.cbrt(G_CubeSumDeviation/(Height*Width));
B_skew=Math.cbrt(B_CubeSumDeviation/(Height*Width));

8. By deriving above methods Color parameters are extracted.

ii. Fuzzy Color Moments

1. Image as input.
2. Extract attributes of the image i.e. Height, Width and RGB values.
3. Use extracted RGB values to calculate color parameters i.e.

   mat[i][j][0] for Red

   mat[i][j][1] for Green

   mat[i][j][2] for Blue
4. Divide the image into 5 regions and calculate three order moments for color parameters.
5. Perform steps 6 to 8 for each of the 5 regions of the image individually.
6. First order moment i.e. means is calculated for R, G and B component of image.

   LOOP i->0 to h-1

   LOOP j->0 to w-1

$$r\_sum = \left(\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} mat[i][j][0] \right)$$

$$g\_sum = \left(\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} mat[i][j][1] \right)$$

$$b\_sum = \left(\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} mat[i][j][2] \right)$$

   here h is the image Height and w is the image Width.

   r_mean=r_sum/(Height*Width)

   g_mean=g_sum(Height*Width)

   b_mean=b_sum/(Height*Width)

7. Second order moment i.e. variance is calculated for R, G and B component of image.

   LOOP i->0 to h

   LOOP j->0 to w

$$R\_SquareSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (mat[i][j][0]-r\_mean)\ \hat{}2$$

h-1 w-1

$$G\_SquareSumDeviation = \sum_{i=0} \sum_{j=0} (mat[i][j][1]-g\_mean)^2$$

$$B\_SquareSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (mat[i][j][2]-b\_mean)^2$$

where h is the Height of the image and w is the Width of the image

R_var = R_SquareSumDeviation / (Height*Width);
G_var = G_SquareSumDeviation / (Height*Width);
B_var = B_SquareSumDeviation / (Height*Width);

8. Third order moment i.e. skewness is calculated for R, G and B component of image.

   LOOP i->0 to h-1

   LOOP j->0 to w-1

$$R\_CubeSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [(mat[i][j][0]-r\_mean)]^3$$

$$G\_CubeSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [(mat[i][j][1]-g\_mean)]^3$$

$$B\_CubeSumDeviation = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [(mat[i][j][2]-b\_mean)]^3$$

Where h is the Height of the image and w is the Width of the image

R_skew=Math.cbrt(R_CubeSumDeviation/(Height*Width));
G_skew=Math.cbrt(G_CubeSumDeviation/(Height*Width));
B_skew=Math.cbrt(B_CubeSumDeviation/(Height*Width));

9. Calculate the overall parameters of the image using the following expression

   Overall parameter = (Central probability * Region 0 parameter) +
   (surrounding region probability * Region 1 parameter) +
   (surrounding region probability * Region 2 parameter) +
   (surrounding region probability * Region 3 parameter) +
   (surrounding region probability * Region 4 parameter)

10. Perform step 9 for each of the three parameter (mean , standard deviation, skewness).

iii. Color Histogram

1. Image as input.
2. Attributes of the image i.e. height, width and RGB values extracted.
3. Color histogram of all images in database extracted.
4. Histogram bins are calculated from RGB matrix
5. First the RGB matrix is normalized
6. HSV from RGB is calculated.
7. Hue, Saturation and value bins are calculated
8. Compare the histogram of query image from histogram of all database images using Euclidean distance metric.

iv. Texture GLCM

1. Image as input.
2. Extract attributes of the image i.e. height, width and RGB values.
3. Co-Occurrence matrix is calculated.
4. Co-Occurrence matrix is normalized.
   using

$$N[i,j] = \frac{P[i,j]}{\sum_i \sum_j P[i,j]}$$

5. Calculate the dissimilarity for Red, Green and Blue component of image.
   LOOP i->0 to n-1
   LOOP j->0 to n-1

$$\text{dissimilarity} = \sum_{i,j=0}^{n-1} P[i,j]\text{mod}[i-j]$$

6. Calculate the mean for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$mean\_i = \sum_{i,j=0}^{n-1} P[i,j]*i$$

$$mean\_j = \sum_{i,j=0}^{n-1} P[i,j]*j$$

7. Calculate the Variance for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$var\_i = \sum_{i,j=0}^{n-1} (i - mean\_i) ^2 * P[i,j]$$

$$var\_j = \sum_{i,j=0}^{n-1} (j - mean\_j) ^2 * P[i,j]$$

8. Calculate the Entropy for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$entropy = \sum_{i,j=0}^{n-1} P[i,j](-logP[i,j])$$

9. Calculate the Angular second moment (energy) for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$asm = \sum_{i,j=0}^{n-1} P[i,j] ^2$$

10. Calculate the Contrast (inertia) for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$con = \sum_{i,j=0}^{n-1} P[i,j](i - j)^2$$

11. Calculate the homogeneity for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$hom = \sum_{i=0}^{n-1} P[i,j] / (1+ (i - j)^2)$$

12. Calculate the standard deviation for Red, Green and Blue component of image.

$$Sd\_i = (var\_i)^{1/2}$$

$$Sd\_j = (var\_j)^{1/2}$$

13. Calculate the Correlation for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$corr = \sum_{i,j=0}^{n-1} (i-mean\_i) * (j-mean\_j) * P[i,j] / (sd\_i*sd\_j)$$

14. By deriving the above methods Texture parameters will be extracted.

v. Moment Invariants

1. Input a Binary Image.
2. Extract attributes of the image i.e. height, width and RGB values.
3. Use extracted RGB values to calculate shape parameters i.e. mat[i][j][3].

Where mat[i][j][0] for Red

mat[i][j][1] for Green

mat[i][j][2] for Blue

4. Calculate the Cartesian product for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$C(p, q) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} i^p * j^q * mat[i][j][3]$$

Where n image dimension

5. Then, calculate the centroid for Red, Green and Blue component of image.

$$\bar{i} = C(1, 0) / C(0, 0) \quad \text{and} \quad \bar{j} = C(0, 1) / C(0, 0)$$

6. Calculate the central moment for Red, Green and Blue component of image.

LOOP i->0 to n-1

LOOP j->0 to n-1

$$mu(p, q) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - \bar{i})^{\wedge}p * (j - \bar{j})^{\wedge}q * mat[i][j][3]$$

7. Calculate the normalized central moment (eta) for Red, Green and Blue component of image.

eta (p, q) = mu (p, q) / (mu (0, 0)) ^gama

Where gama= (p+q)/2+1

8. Calculate the seven invariant moments (phi) for Red, Green and Blue component of image**.**

phi1 = eta (2,0) + eta (0,2)

phi2 = (eta (2,0) – eta (0,2)) ^2 + 4 * eta (1,1) ^2

phi3 = (eta (3,0) – 3 * eta (1,2)) ^2 + (eta (0,3) – 3 * eta (2,1)) ^2

phi4 = (eta (3,0) + eta (1,2)) ^2+ (eta (0,3) + eta (2,1)) ^2

phi5 = (3 * eta (3,0) – 3 * eta (1,2)) (eta (3,0) + eta (1,2)) [(eta (3,0) +eta (1,2)) ^2– 3(eta (2,1) + eta (0,3)) ^2] + (3 * eta (2,1) – eta (0,3)) (eta (2,1) +eta (0,3))× [3(eta (3,0) + eta (1,2)) ^2 – (eta (2,1) + eta (0,3))^2]

phi6 = (eta (2,0) – eta (0,2)) [(eta (3,0) + eta (1,2)) ^2 – (eta (2,1) + eta (0,3)) ^2] + 4 eta (1,1) (eta (3,0) + eta (1,2)) (eta (2,1) + eta (0,3))

phi7 = (3 * eta (2,1) – eta (0,3)) (eta (3,0) + eta (1,2)) [(eta (3,0) + eta (1,2)) ^2-3(eta (2,1) +eta (0,3)) ^2] + (3 * eta (1,2) – eta (3,0) (eta (2,1) +eta (0,3)) × [3(eta (3, 0) + eta (1, 2)) ^2 – (eta (2, 1) + eta (3, 0)) ^2]

9. By deriving the above method Shape parameters will be extracted.

3.5.2 Pattern Recognition

i. ObjectBasedRecognition

1) Input is given in the form of 3D RGB Matrix.
2) Required Query Object is selected.
3) Technique is selected
4) Object parameters are calculated according to the selected technique

    4.1) If technique selected= "colormoments"

    Objectparameters =call algorithm colormoments

    4.2) If technique selected= "colorhistogram"

    Object parameters =call algorithm colorhistogram

    4.3) If technique selected= "textureglcm"

    Object parameters =call algorithm textureglcm

    4.4) If technique selected= "fuzzycolor moments"

    Object parameters =call algorithm fuzzy color moments

    4.5) If technique selected = "gaborfilter"

    Object parameters = call algorithm gabor filter

    4.6) If technique selected = "momentinvariant"

    Object parameters = call algorithm moment invariants

5) Parameters of all the templates of the input image are fetched.
6) The distance of the parameters of all templates and the object parameters are computed
7) N minimum templates are highlighted on the result image. The value N is selected by the user.(N<7)
8) Result can be saved to the desired location.

ii. Target Based Recognition

1) Input is given in the form of 3D RGB Matrix.
2) Technique selected
3) SourceImageParameters are calculated according to the selected technique

    3.1) If technique selected= "colormoments"

    SourceImageParameters =call algorithm colormoments

3.2) If technique selected= “colorhistogram”

SourceImageParameters =call algorithm colorhistogram

3.3) If technique selected= “textureglcm”

SourceImageParameters =call algorithm textureglcm

3.4) If technique selected= “fuzzy color moments”

SourceImageparameters =call algorithm fuzzy color moments

3.5) If technique selected = “gabor filter”

SourceImageParameters = call algorithm gabor filter

3.6) If technique selected = “moment invariant”

SourceImageParameters = call algorithm moment invariants

4) Parameters of all the database target images are fetched.
5) The distance of the parameters of all the database images with that of the input image are computed
6) Minimum 3 distances are considered and the corresponding images appear on the screen
7) Results can be saved to the desired location.

The various algorithms for importing and deleting images for the two pattern recognition applications are presented henceforth.

Import Image

1) Image path of the image to be imported is the input

2) Image is checked for the correct format and size

2.1 If no

Image rejected
return

3) The parameters according to all the techniques calculated.

Parameters= call algorithm colormoments

Call algorithm colorhistogram

Call algorithm textureglcm

Call algorithm fuzzy color moments

Call algorithm gabor filter

Call algorithm moment invariants

4) Calculated parameters stored in the database.

Delete Image

1) Image path of the image to be deleted is the input

2) Image is checked if present in the database

2.1) If no

Error message generated
return

3) Information related with the image is deleted from the database.

3.6 Code Efficiency

Efficiency creation of code provides for better working of program, improves reliability as the code will be up to the mark and hence easier debug. So to improve the code efficiency the following steps have to be taken:-

1 Our code is well documented since we have provided explanation for each and every operation being performed.

2 There is no dead code in this project.

The code that is frequently used throughout the program is modularized.

Then certain variable that are used throughout the software have are declared as global.

The code is free from any bug.

3.7 Validation Checks

To make the software more users friendly and to prevent the user from entering garbage values, the following constraints and event have been applied:

1. If user inputs any image except from bmp then program will give error.
2. Buttons are enabled and visible according to the application selected.
3. If database is absent it is created first.
4. In case of Object Based feature Extraction input image has to be from DatabaseFEImage folder otherwise error is generated.

5. In case of import and delete system asks for confirmation.

3.8 Certain issues raised in development

Problem: Java heap error: in case of certain big images the system was showing this error

Solution:"- Xmx1024m" was added to the VMOptions of the project.

CHAPTER 4

RESULTS & ANALYSIS

CHAPTER 4
RESULTS & ANALYSIS

Some Assumptions and Facts:

Input Image Size: 659 x 449

Query Object Image Size: 50 x 50

Distance between two templates: 5 pixels

No of templates to match in the input image: 9760

Similarity Comparison Technique followed: Euclidean Distance Measuring

The following graphs show the distance variation in the top 18 results in the various feature extraction techniques.

Distance comparison for color moments

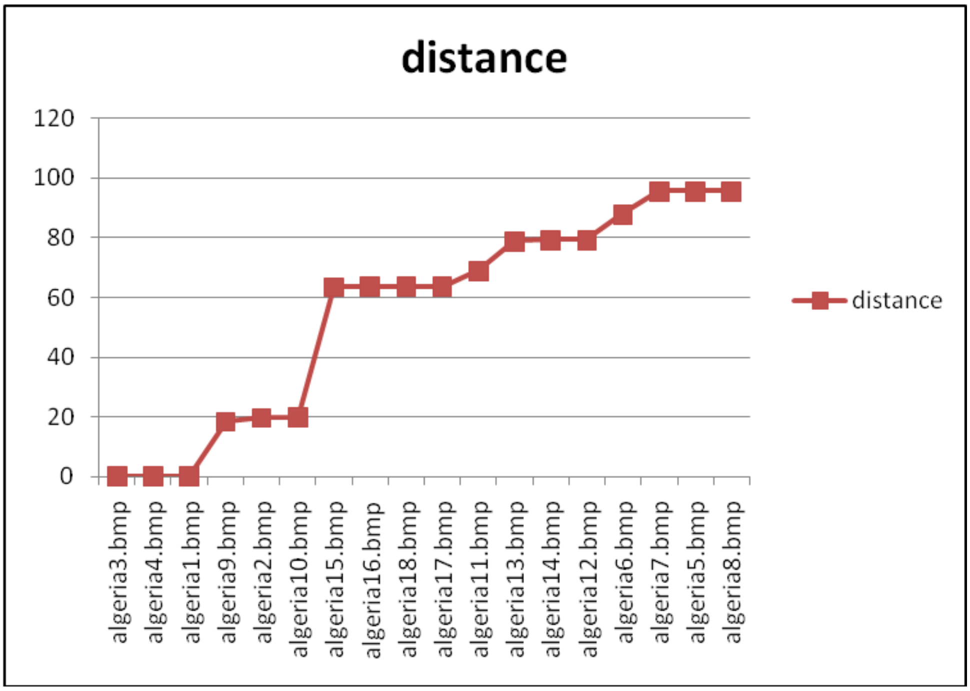


Figure 4.1: Distance Comparison graph for color moments.x axis represents the instances of the different templates in the best 18 results. Y axis represents the distance value between the input image and the images in the database.

The above graph shows that the variation in the distance value is low initially when the various transformed images are compared. But the variation increases if the images are made color variant.

Distance comparison for fuzzy color moments

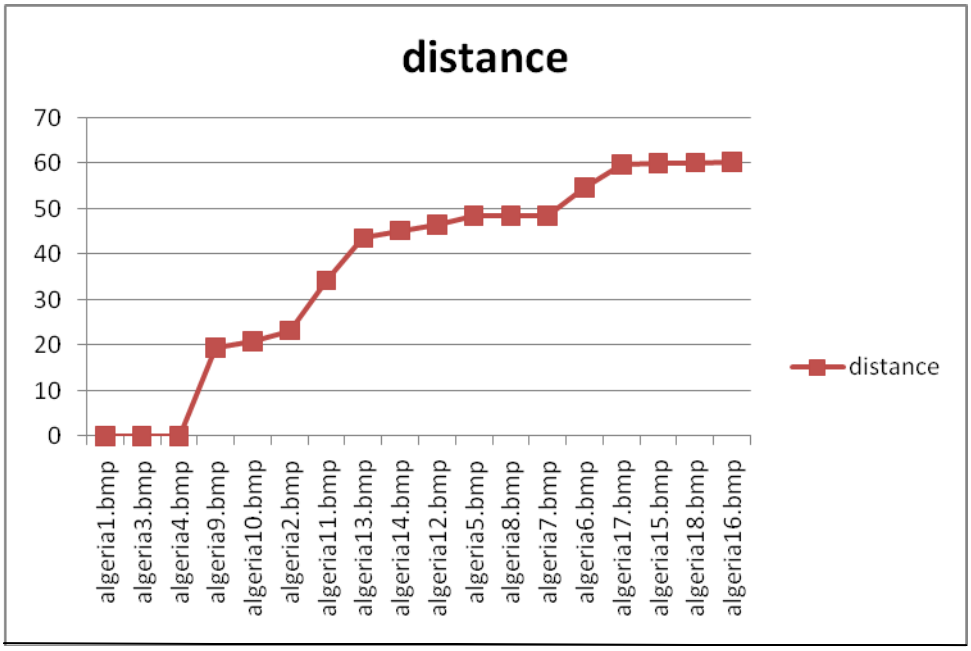


Figure 4.2: Distance Comparison graph for fuzzy color moments parameters. x axis represents the instances of the different templates in the best 18 results. Y axis represents the distance value between the input image and the images in the database.

Figure 4.2 shows that the variation in the distance parameter is less initially as in the case of the color moment parameters. However, unlike color moments the variation is slightly more uniform even when the images in the database are transformed with respect to both dimension and color.

Distance comparison for color histogram

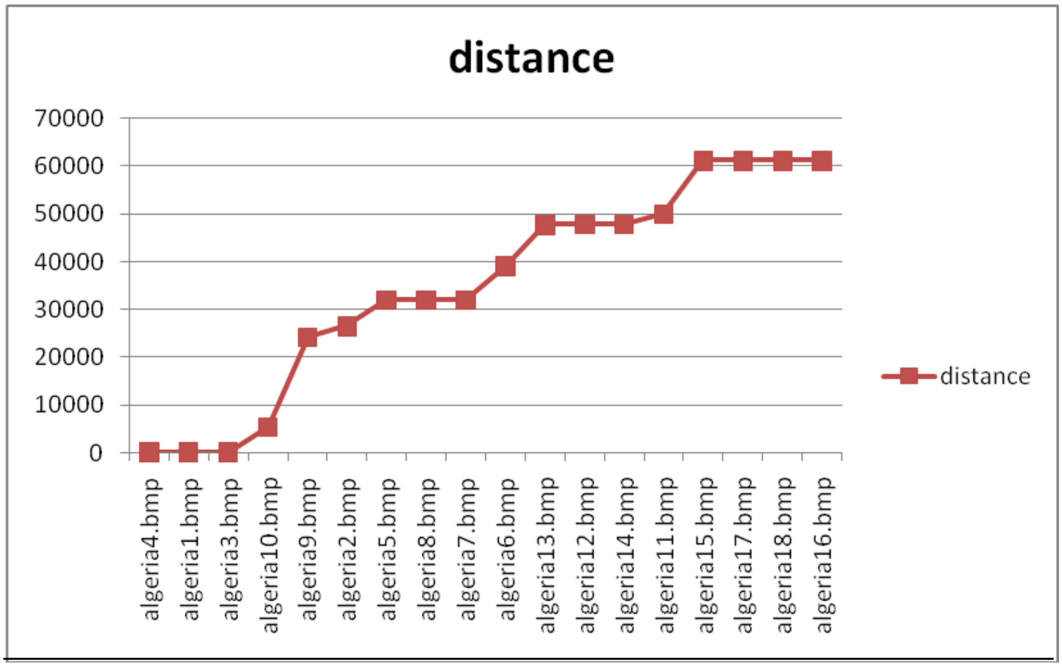


Figure 4.3: Distance Comparison graph for color histogram parameters. X axis represents the instances of the different templates in the best 18 results. Y axis represents the distance value between the input image and the images in the database.

The distance varies more steeply in case of the color histogram feature parameters. The color histogram parameters however show little variation initially and in the images transformed by rotation. The images transformed due to skewness however show major variation in the distance.

Distance comparison for texture glcm

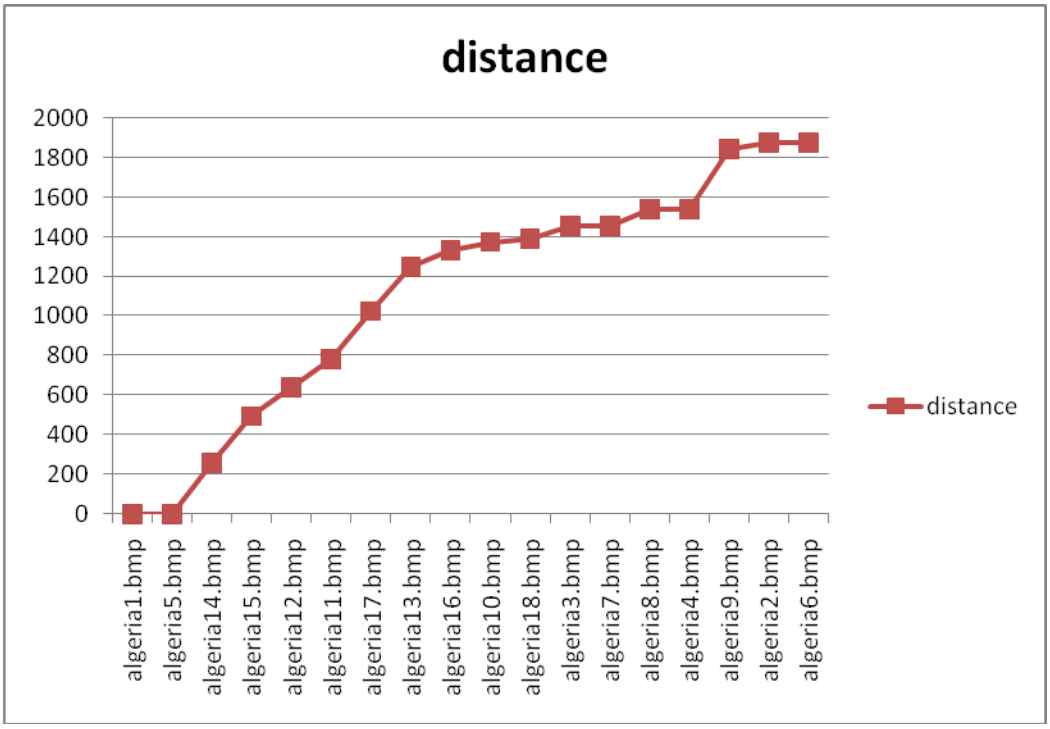


Figure 4.4. Distance Comparison graph for grey level co-occurrence matrix (GLCM) parameters. x axis represents the instances of the different templates in the best 18 results. Y axis represents the distance value between the input image and the images in the database.

The distance variation increases very steeply in case of the texture glcm. The distance variation can be assumed to closely follow a linear pattern.

Distance comparison for gabor filter

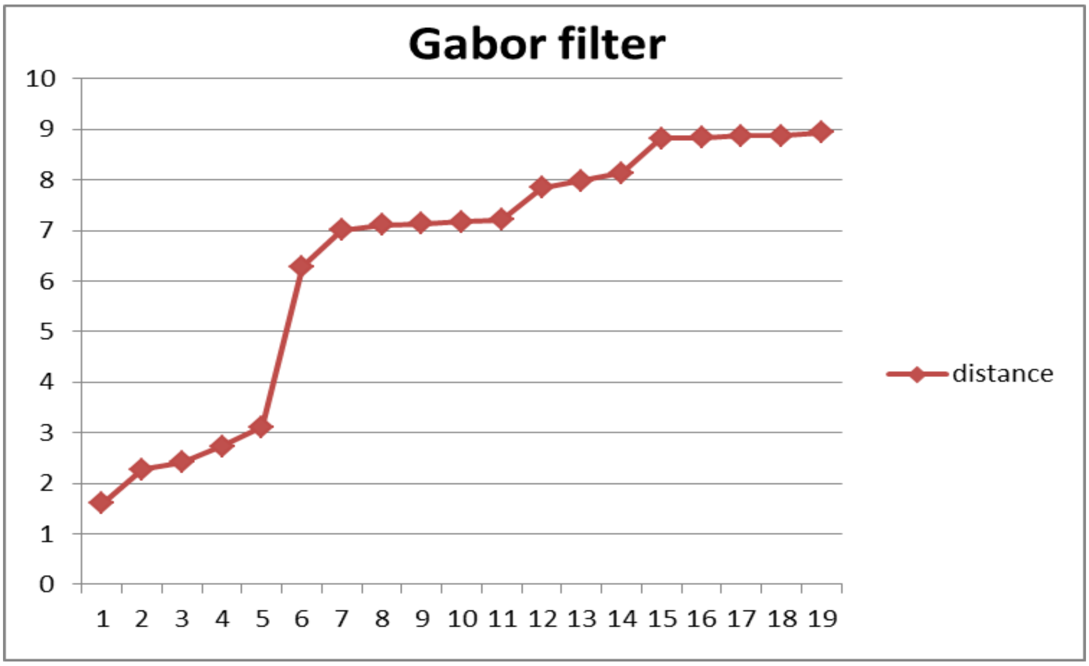


Figure 4.5: Distance Comparison graph for gabor filter parameters. x axis represents the instances of the different templates in the best 20 results. Y axis represents the distance value between the input image and the images in the database.

The distance variation in the gabor filter is slow in the beginning, but it increases steeply when a certain variation in the images occur, after which the distance varies gradually.

Distance comparison for moment invariant

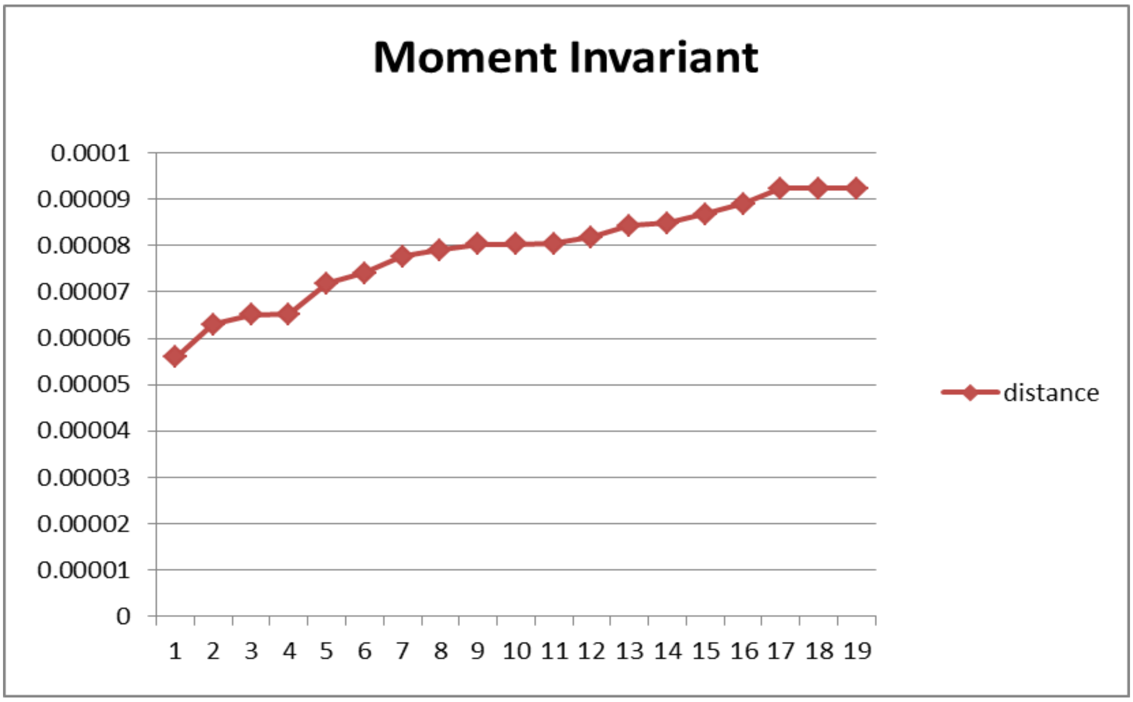


Figure 4.6. Distance Comparison graph for moment invariant parameters. x axis represents the instances of the different templates in the best 20 results. Y axis represents the distance value between the input image and the images in the database.

Moment invariant is a shape based feature extraction technique. It shows a gradual change in the distance between the feature parameters in the best 20 results in pattern recognition system.

Comparison for time analysis for various techniques

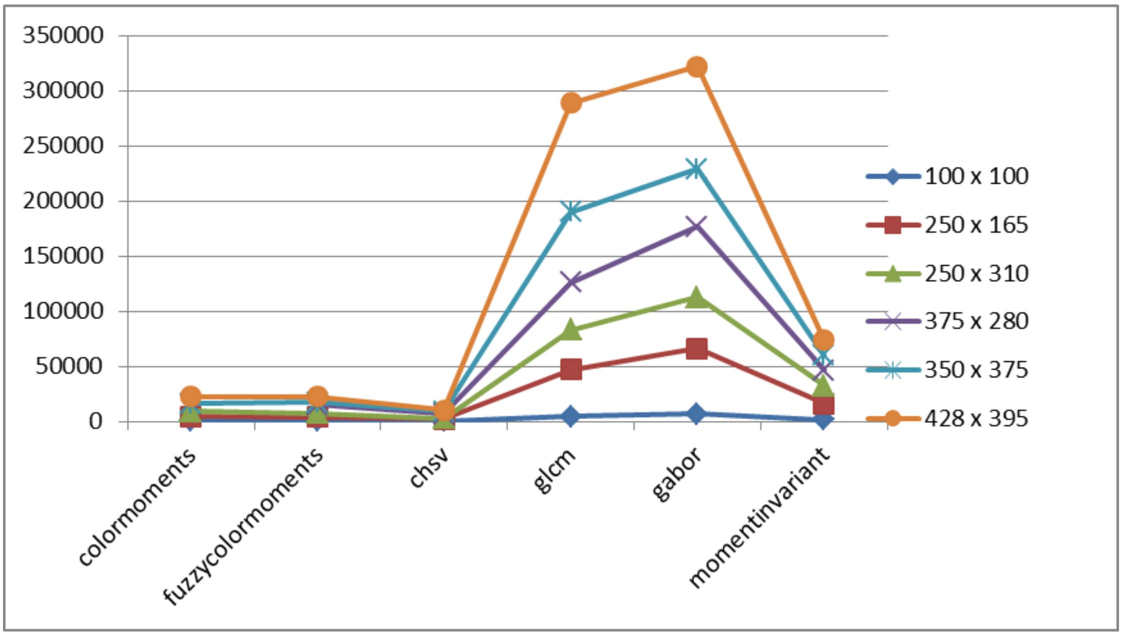


Figure 4.7. Variation for the time taken in importing feature parameters for various images of different sizes. X axis represents the different techniques used for feature extraction. Yaxis represents the time value (in msec). The different line graphs represents the different image sizes.

The time taken to calculate the various feature parameters increases as the size of the image increases. The time taken to calculate color parameters is less, while it is higher to calculate the shape features. The time increases steeply in case of the texture parameters.

Since the time taken tocalculate the color features is very small and its variation is not very clear in the Figure 4.7; the Figure 4.8 shows the variation in calculationg the color parameters.

Comparison for time analysis for color techniques

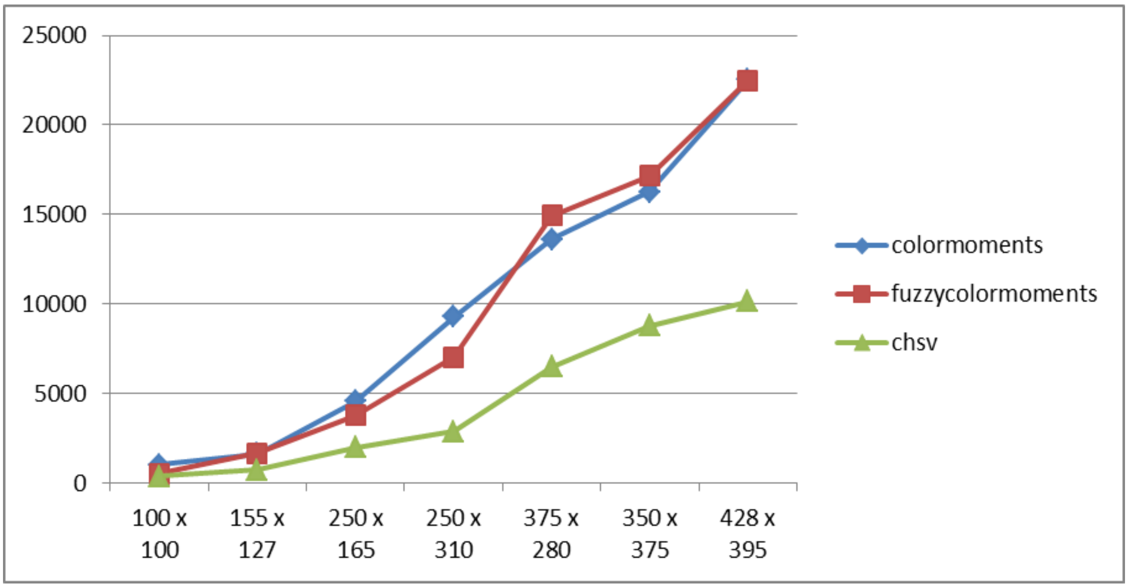


Figure 4.8: Variation for the time taken in importing color feature parameters for various images of different sizes. X axis represents the different image sizes. Yaxis represents the time value (in msec). The different line graphs represents the color techniques used for feature extraction.

The time to calculate the various color parameters is almost comparable when the image size is small. However, it increases manifolds as the image size increases. The rate of increase in the time taken to calculate the feature parameters increases steeply as the size of the input image increases.

Comparison of accuracy percentage of pattern recognition for different techniques

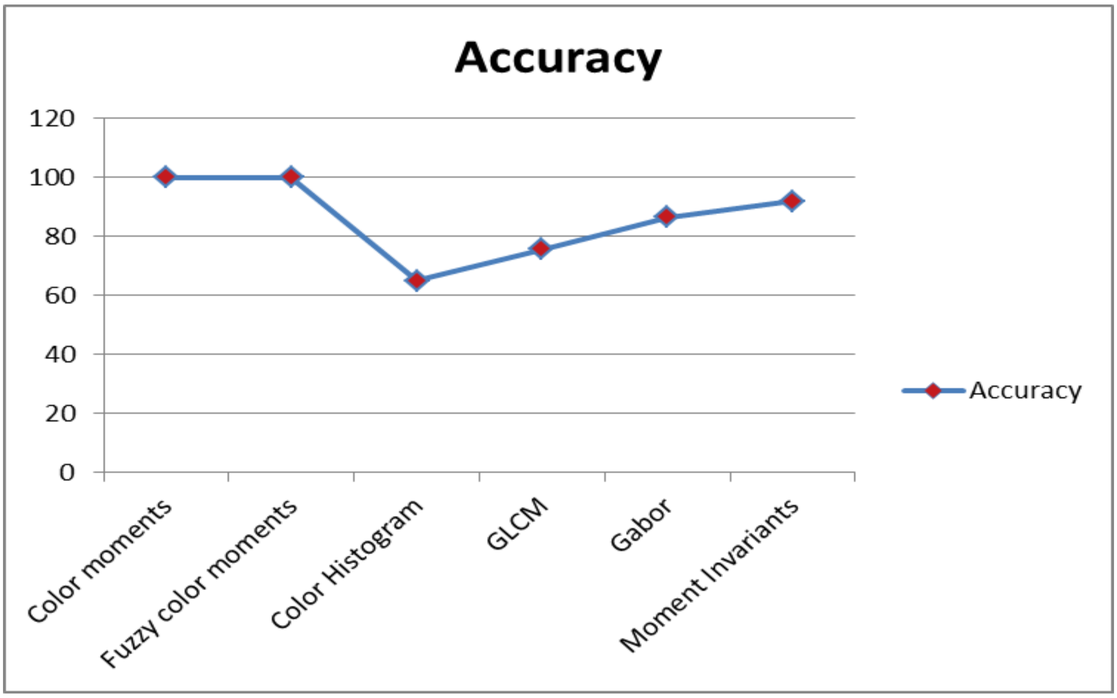


Figure 4.9: Variation of accuracy percentage of various feature extraction techniques w.r.t. pattern recognition

The Figure 4.9 shows the accuracy percentage in case of pattern recognition done through the various feature extraction techniques. A color and shape prominent image when used as an input shows the above variation. The color moments and fuzzy color moments identify all of the required object templates, but they also mark some extra redundant templates to be a part of the result. The moment invariant technique however used identifies almost all the objects, except a few. It yields an accuracy percentage of about 91.6%. Also the moment invariants technique show fewer redundant results as compared to the color moments and fuzzy color moments techniques.

Comparison of various feature extraction techniques

Techniques	No. Of features	Rotation	Skewness	Inversion of Color	Standard Deviation
Color Moments	9	Invariant	Partially Invariant	Variant	35.26484684
Fuzzy Color Moments	9	Invariant	Partially Invariant	Variant	21.56509376
Color Histogram	24	Invariant	Partially Invariant	Variant	21.56509376
Grey Level Co-Occurrence Matrix	18	Variant	Variant	Invariant	35.26484684
Gabor Filter	12	Variant	Partially Variant	Invariant	75.8621474
Moment Invariant	21	Invariant	Partially Variant	Invariant	27.654913

Table 4.1: Comparison of various feature extraction techniques

CONCLUSION & FUTURE WORK

CONCLUSION

The content-based image retrieval is an interesting and complex problem studied by many researchers all over the world. The complexity of this problem is due to many factors. The most important among them are a “semantic gap” between semantic content of the image and its visual features, and necessity to search large amount of multidimensional data in real time. In this project, few popular algorithms for constructing feature vectors that reflect various image features, such as color, texture, and shape of the objects have been discussed. The following results have been observed.

- Color based approaches are invariant to rotation, and are partially affected by skewness. They are however, greatly affected by an inversion of color component values.
- On contrast, Texture based approach GLCM is affected by rotation and skewness but is invariant to inversion of color.
- Color Histogram is computationally least inexpensive. While texture glcm is most expensive.
- Fuzzy color moments do not produce wide variation in results in case of satellite images. It however gives better results as compared to color moments in case of images with objects.
- Texture GLCM gives better results in analysis of images in case of satellite images, in case of terrain and landform analysis.
- The gabor filter though computationally expensive but identifies the texture characteristics very effectively. It captures the orientation of the texture.
- The shape based moment invariant technique is very effective in retrieving shape based patterns.
- The decision tree algorithm is used to generate decision about the scene represented in the image.

FUTURE WORK

CBIR is a vast field with a large number of researchers working on it. The success of a CBIR system lies in the fact how efficiently the features of an image can be extracted. This project can be further extended to retrieve the objects by using a combination of different types of features.

The Image Understanding process combines the various results of object identification and depending on the decision generating algorithms performs scene analysis.The Image Understanding system generated is applicable to specific case of airbase. It can be developed to support a more generic system. The expectation values used for analysis are based on research work; it can be developed to understand the images of a varied environment instead of a fixed case.

REFERENCES

[1]. A. Jain, N. Ratha, S. Lakshmanan, "Object detection using Gabor filters", In Pattern Recognition, vol. 30, pp. 295-309, 1997.

[2]. A. K. Jain, and F. Farroknia, "Unsupervised texture segmentation using Gabor filters," *PatternRecognition*, Vol.24, No.12, pp. 1167-1186, 1991.

[3]. A. Sajjanhar, and Lu, G., A Grid Based Shape Indexing and Retrieval Method, *Australian Comput. J.,* 1997, vol. 29, issue 4, pp. 131–140.

[4]. B.S. Manjunath and W. Y. Ma, " Tecture Features for browsing and Retrieval of Image Data", IEEE Transaction on Pattern Analysis and Machine Learning, Vol. 18, No. 8, Aug, 1996

[5]. Etienne Barnard and David Casasent, "Image Processing For Image Understanding With Neural Nets"

[6]. Feng Shaorong:Research and Improvement of Decision Trees Algorithm .Journal of XiamenUniversity (NaturalScience), 2007,pp.498-500.

[7]. H. Zareipour, Student Member, IEEE, K. Bhattacharya, Senior Member, IEEE, C.A. Ca˜nizares, Senior Member, IEEE, "Forecasting the Hourly Ontario Energy Price by Multivariate Adaptive Regression Splines

[8]. Hu, M.-K., Visual Pattern Recognition by Moment Invariants, *IEEE Trans. Information Theory*, 1962, vol. 8, issue 2, pp. 179–187.

[9]. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer graphics: principles and practice,* 2nd ed., Reading, Mass, Addison-Wesley, 1990.

[10]. John Shunen Shieh, “Relational Fuzzy Model: A Representation For Object Identification”, Proceedings of the 1993 IEEE, /RSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 2630,1993

[11]. KoSir, Andrej*, TasiE, Jurij F.*, “Formal system based on fuzzy logic applied to digital image scene analysis”

[12]. L. Yang, and F. Algregtsen, "Fast computation of invariant geometric moments: A new method giving correct results," *Proc. IEEE Int. Conf. on Image Processing*, 1994.

[13]. M. K. Hu, "Visual pattern recognition by moment invariants," in J. K. Aggarwal, R. O. Duda, and A.Rosenfeld, *Computer Methods in Image Analysis*, IEEE computer Society, Los Angeles, CA, 1977.

[14]. M. Stricker, and M. Orengo, "Similarity of color images," *SPIE Storage and Retrieval for Image and Video Databases III*, vol. 2185, pp.381-392, Feb. 1995

[15]. Miao Zhenjiang, Yuan Baozong, “A NN Image Understanding System for Maps and Animals Recognition”, at IEEE Tencon, ’93, Beijing

[16]. Mobasher B, Cooley R, Jaideep S, etal, “Comments on decision tree”, New York:IEEE Press,1999.

[17]. N.S.Vaselier, “Content-based Image Retrieval Methods”, Programming and Computer Software,2009, Vol.35, No.3, pp. 158-180

[18]. P. Brodatz, "Textures: A photographic album for artists & designers," Dover, NY, 1966

[19]. P.S.Hiremath, S. Shivashankar, and Jagadeesh Pujari, “Wavelet Based Features for Color Texture Classification with Application to CBIR”. International Journal of Computer Science and Network Security, Vol. 6 No. 9A, Sept, 2006

[20]. Qiang Hou, HePing Pan, Juan Li, Ti Wu, “Natural Image Understanding via Sparse Coding”, DOI: 978-1-4244-5824-0/$26.00 _c 2010 IEEE

[21]. Quinlan J R, “Induction of decision tree”, Machine Learning,1986, (1),pp.81-106

[22]. Stricker, M. and Dimai, A., Spectral Covariance and Fuzzy Regions for Image Indexing, *Machine Vision Applications*, 1997, vol. 10, pp. 66–73.

[23]. Sun Chaili: Study on Data Stream M in ing Algorithm Based on Decision Tree Journal of Taiyuan University of Science and Technology.,2006, pp.269-270.

[24]. T. Barbu, “Novel automatic video cut detection technique using Gabor filtering”, Computer and Electrical Engineering, DOI:10.1016/j.compeleceng.2009.02.003.

[25]. T Tjahjadl and R Henson, “A Knowledge Based System For Image Understanding”

[26]. Tudor Barbu, “Content–based Image Retrieval using Gabor Filtering” in 20th International Workshop on Database and Expert Systems Application, 2009

[27]. Vidya Manian and Marcel J. Castro, “ Color Image Classification using Gabor Filters”, Computing Research Conference Proceedings, May, 2000

APPENDIX

APPENDIX 1: IMPLEMENTATION DETAILS

NOMENCLATURE

Following are the nomenclature used in the development of the software

- **Database**

DatabaseFEImage—the source image used in the application in object based feature extraction

- DatabaseTargetImage—the source image used in the application in target based feature extraction
- QueryObjectImage—These are the objects to be identified in the application object based feature extraction

- **Objects of different classes used in CMainFrame.java**

Class	Object
CParameter	ColorParameter
Texture_GLCM	GLCM
CImageFormats	ptCImageFormats
CUtility	Utility
CHSV	ColorHistogram
CGabor	obGabor
CMomentInvariant	obMomentInvariant
CImageAnalysis	obImageAnalysis
CShortestDistance	obShortestDistance

Global Variable names

Variable Name	Variable Description	Variable Type
SourceImage	input image selected by the user	An object of class Image
ROSourceImage	image in the rendered form	An object of class RenderedOp
SourceImagePath	Path of the input image	An object of class String
SourceImageDataMatrix	RGB matrix of the input bmp image	3D int matrix
SourceImageHeight	height of the input image selected by the user	Int
SourceImageWidth	width of the input image selected by the user	Int
ResultImage	resultant image which is the output according to the different applications and techniques.	An object of class Image
ResultImagePath	Path of the input image	An object of class String
ROResultImage	image in the rendered form	An object of class RenderedOp
ResultImageDataMatrix	RGBmatrix of the output bmp image	3D int matrix
ResultImageHeight	height of the ouput image selected by the user	Int
ResultImageWidth	width of the output image selected by the user	Int
QueryObjectImage	query objects present in the database.	An object of Class Image
QueryObjectImagePath	Path of the query object image	String
TargetImageHeight	height of the target image selected by the user.	Int

TargetImageWidth	width of the target image selected by the user.	Int
QueryObjectImageHeight	height of the QueryObjectImage	Int
QueryObjectImageWidth	width of the QueryObjectImage	Int
QueryObjectImageDataMatrix	matrix of the QueryObjectImage	3D int matrix

PROCESS DESCRIPTIONS

Function: Object Based Action Performed

Input	Source Image from user
Output	• Display query objects on the "QueryObjectInternalFrame" which is on "QueryObjectPanel" • Display SourceImage selected by the user on "SourceImageLabel" which is on "SourceImageInternalFrame"

Function: Target Based Action Performed

Input	SourceImagePath
Output	SourceImage selected by the user on "SourceImagePanel" which is on "SourceImageInternalFrame".

Function : ImportDatabaseFEImageActionPerformed()

Input	ImportDatabaseFEImagePath from the user
Output	Status of import whether successful or not.

Function : ImportDatabaseTargetImageActionPerformed

Input	ImportDatabaseTargetImagePath from user
Output	Status of import whether successful or not

Function : ImportQueryObjectImageActionPerformed()

Input	ImportImportQueryObjectImagePath from the user
Output	Status of import whether successful or not

Function: Delete DatabaseFEImage

Input	bmp image selected by the user
Output	bmp image selected is deleted

Function: DeleteTargetImageActionPerformed()

Input	bmp image selected by the user
Output	bmp image selected is deleted

Function: DeleteQueryObjectActionPerformed()

Input	QueryObject selected by the user
Output	selected QueryObject is deleted

Function: Delete Database

Input	Selection by the user
Output	Database deleted

Function : AddImageToDatabase()

Input	• ImageHeight: int • ImageWidth : int • ImageDataMatrix : int 3D RGBmatrix • ImageName : String • TechniqueName : String
Output	ImportImageName : String if the image is renamed the changed name is returned else the same name is returned

Function : StoreTemplateToDatabase()

Input	• ImageHeight: int • ImageWidth : int • ImageDataMatrix : int 3D RGBmatrix • ImageName : String • TechniqueName : String
Output	All the parameters of all the templates of the image are stored in the database

Function: GetQueryObjectResult

Input	• SourceImagePath: String • SelectedQueryObjectImagePath : String • TechniqueName : String
Output	• ResultImage : Image

SOME MORE FUNCTIONS USED

- **void ConnectToDatabase( )**
  - connects to database by first reading the password from "SQLPassword.txt". If the password and username doesn't authenticate ,new password and username are asked from the user.
  - If the database doesn't exist ,new database is created by building a connection to database using the default database.

- **void DisplayQueryObjectImage( )**
  - Reads the queryobjects from the database table "QueryObjectName_Table" and the image from the folder "QueryObjects".
  - Displays all the queryobjects on "QueryObjectInternalFrame"

- **double[] CalculateImageColorParameters(String ImagePath)**
  - calculates the color moments of the image at the ImagePath by first forming the 3D RGB datamatrix. Color Parameters are calculated using ColorParameter.fn_getColorParameters(ROImage.getHeight(),ROImage.getWidth( ),ImageDataMatrix);

- **void StoreTemplateToDatabase(int ImageHeight,int ImageWidth,int ImageDataMatrix[][][],String ImageName, String TechniqueName)**
  - stores the parameter values o all the technique of all the tenplates formed by the image

- **Image HighlightMatchedTemplate(double threshold,int NoOfResults)**
  - highlights the closest matching template on the basis of threshold and returns the ResultImage formed thereby.

- **Image GetTargetImageResult(String SourceImagePath)**
  - computes the closest matched image by creating a temporary table databasetarget_source_distance which stores the distance of each

databaseTargetImage from the SourceImage and returns the Image with the minimum distance.

- **void initializeMainFrame()**
  - Does the initial task of making all the GUI components not visible and initializing certain flags.

- **void Create Database()**
  - creates the database “PRISM” using the default database. Also creates the initial tables required in the database.
    - Databasefeimagename_table
    - databasetargetimagename_table
    - queryobjectname_table
    - targetimagecolorhistogram
    - targetimagecolormoments
    - targetimagetextureglcm

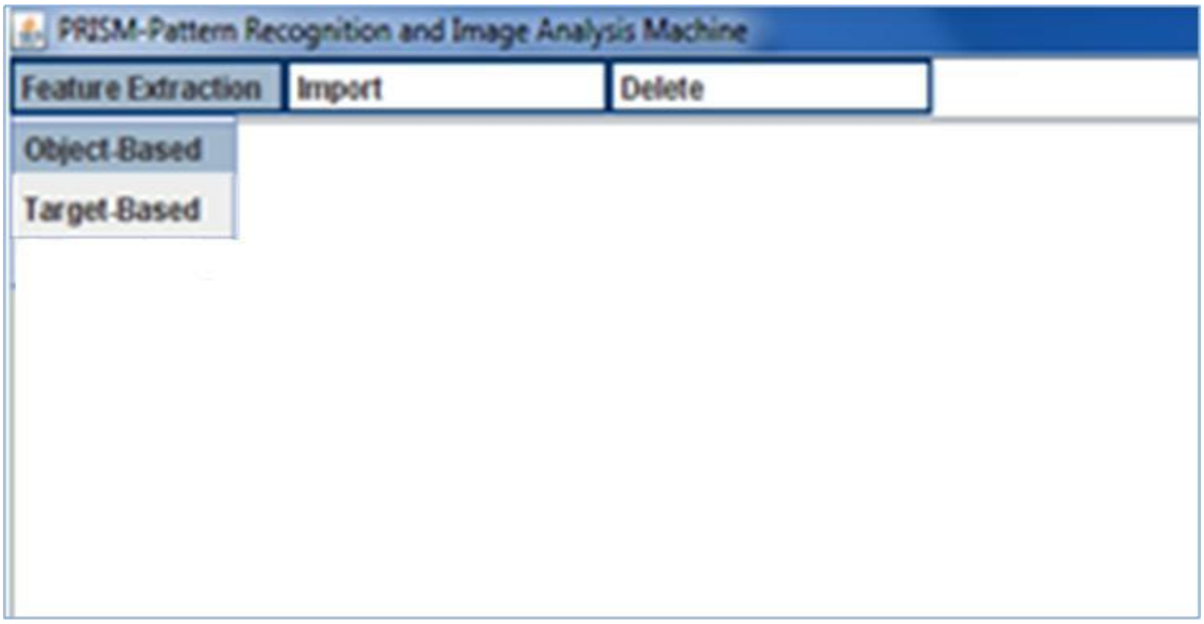
- **void ProceedButtonActionPerformed()**
  - Calls the result computing functions on the basis of 3 different Applications viz. “ObjectBased” ,”TargetBased”, “Image Analysis”
  - Displays the result on ResultImageInternalFrame.

- **void TechniqueComboBoxActionPerformed()**
  - Action Peformed for TechniqueComboBox
  - Sets the TechniqueSelectedFlag to 1

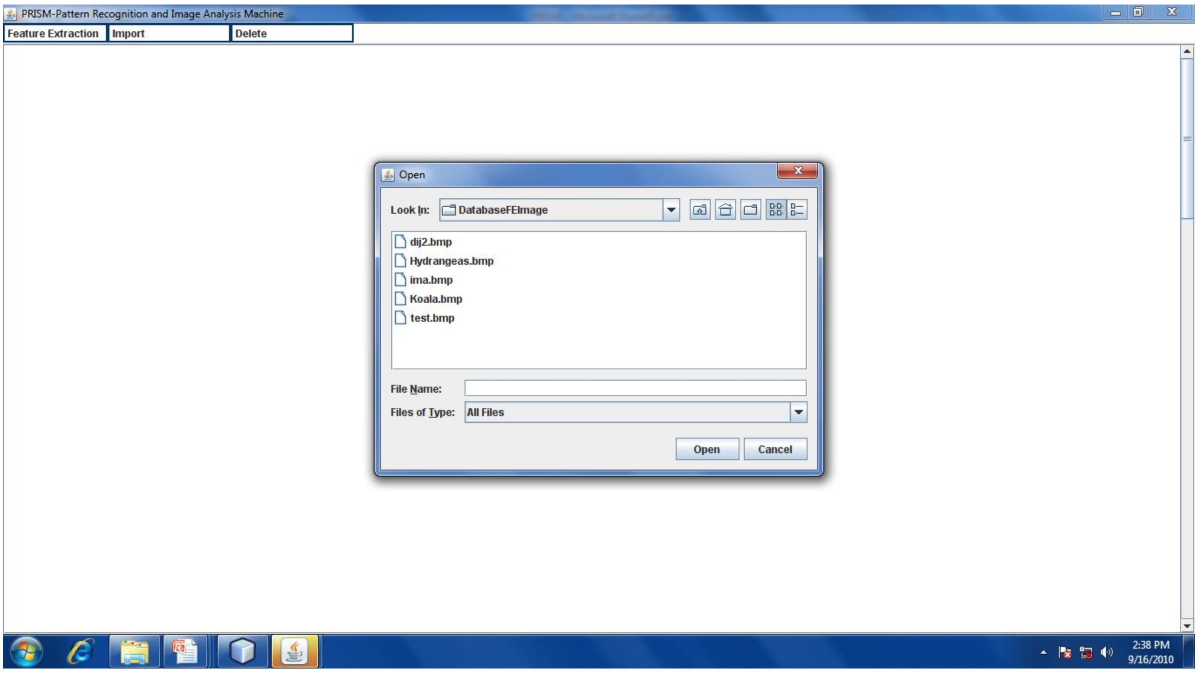
APPENDIX 2: SCREEN LAYOUTS

Object Based Recognition:

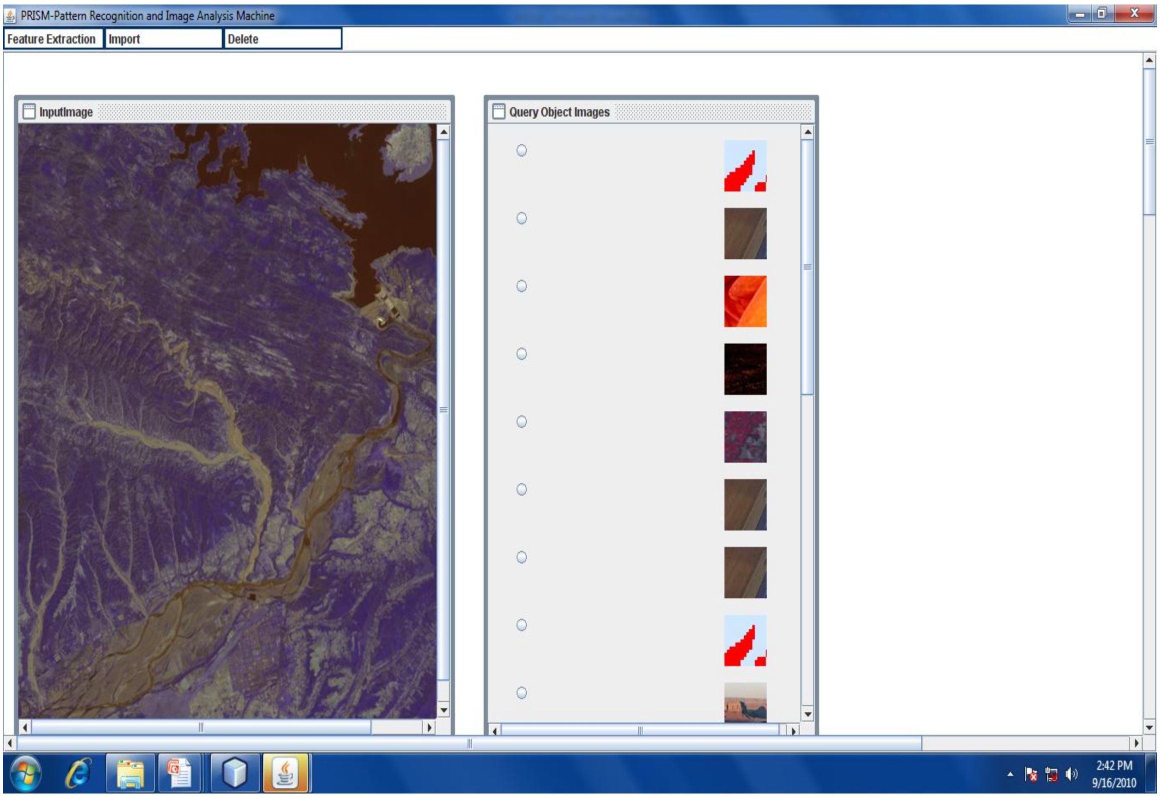
Displays the selection of Object Based Feature Extraction



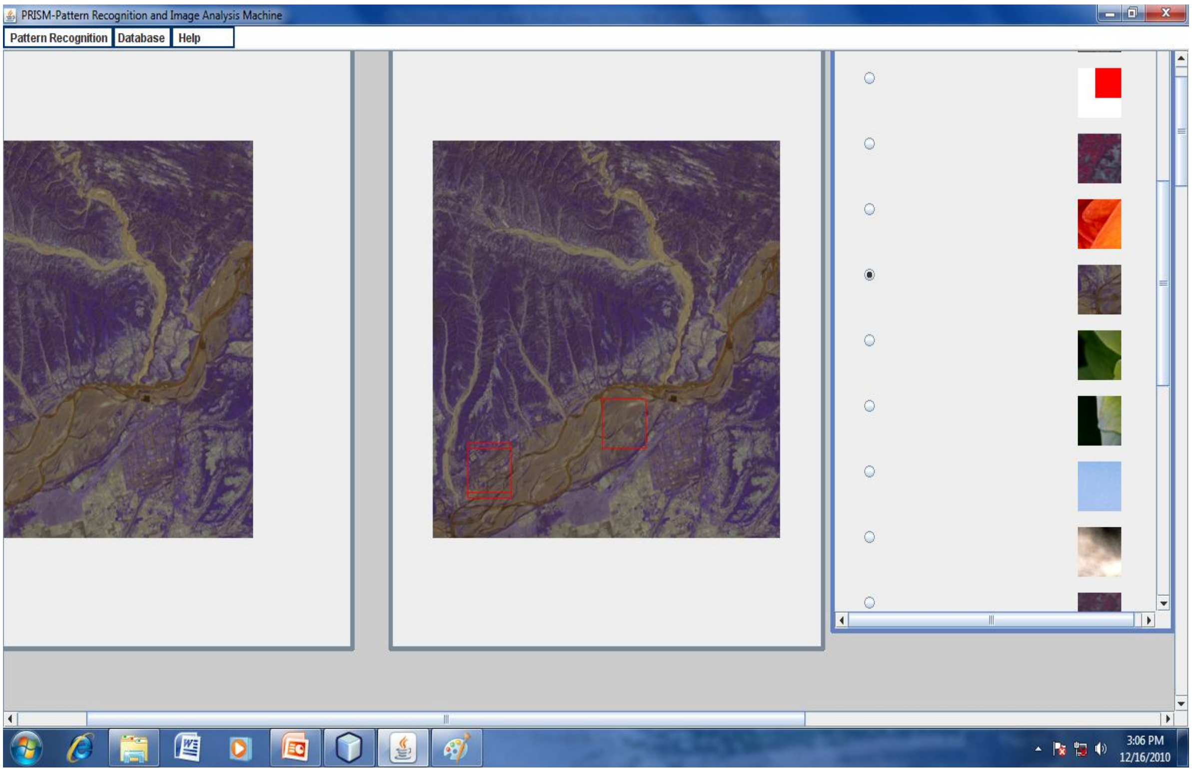
On selection of object based File Chooser opens up



After selection of an image the image as well as query objects are displayed



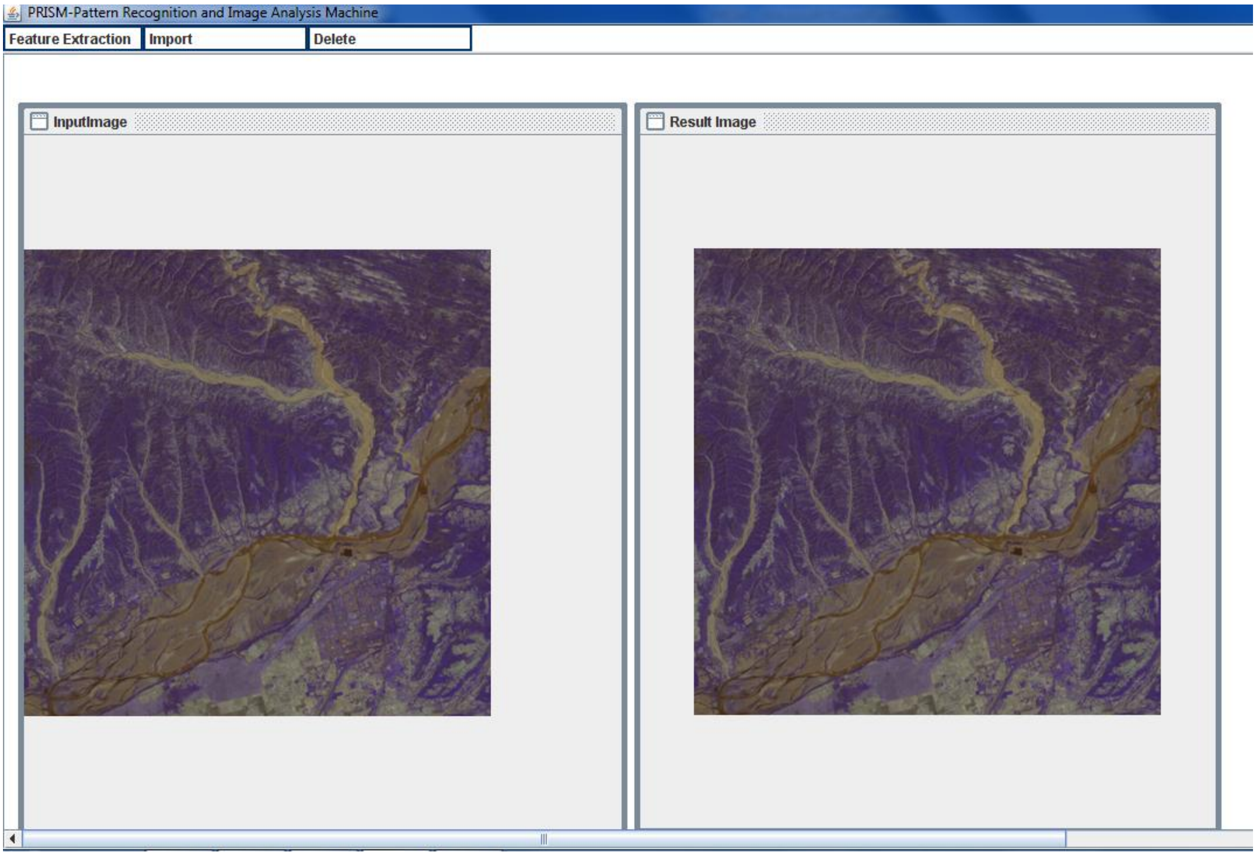
A particular technique is selected and Proceed button is clicked to activate the processing.



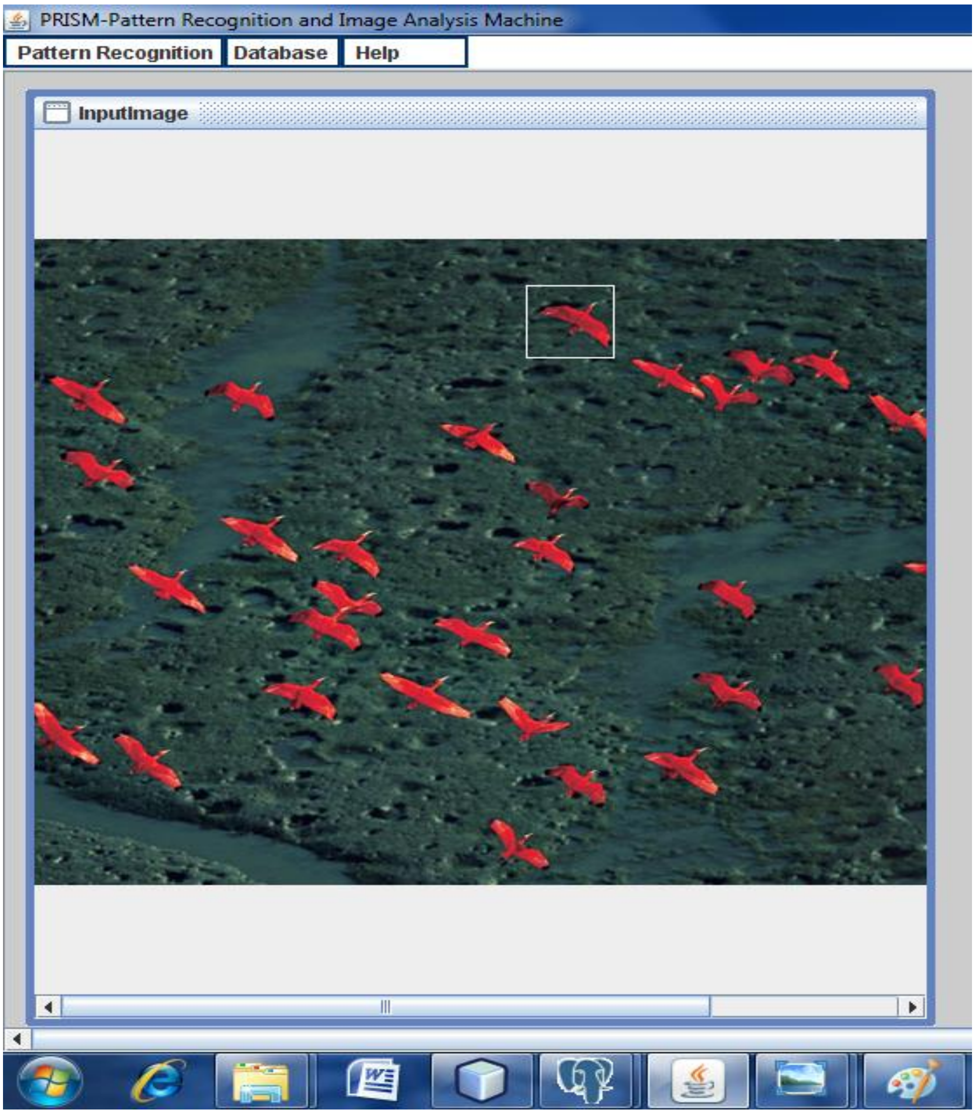
Target Based Recognition

After selection of the application and the input image the input image appears on the screen.

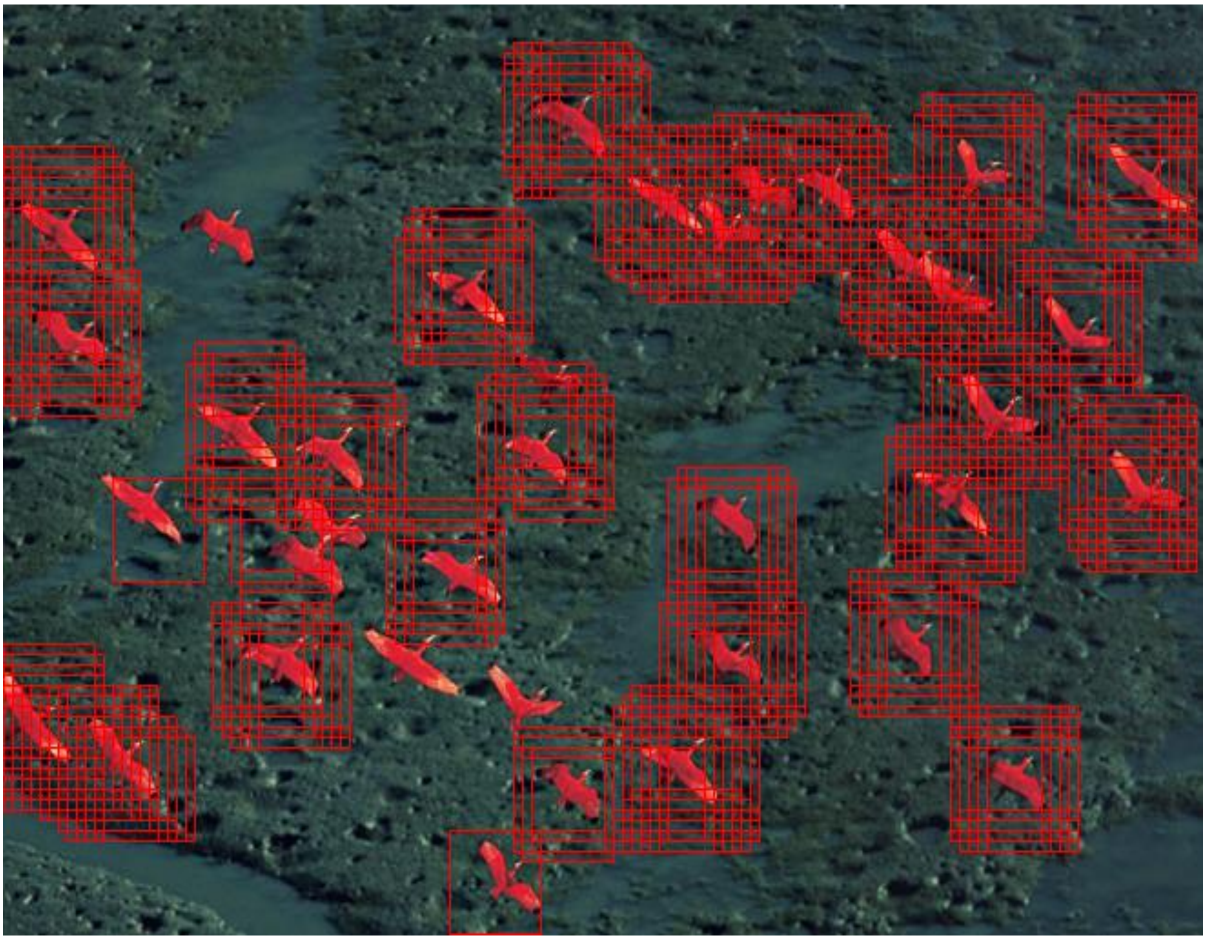
After selection of a technique and clicking of proceed button top 5 Result Images appears on the screen



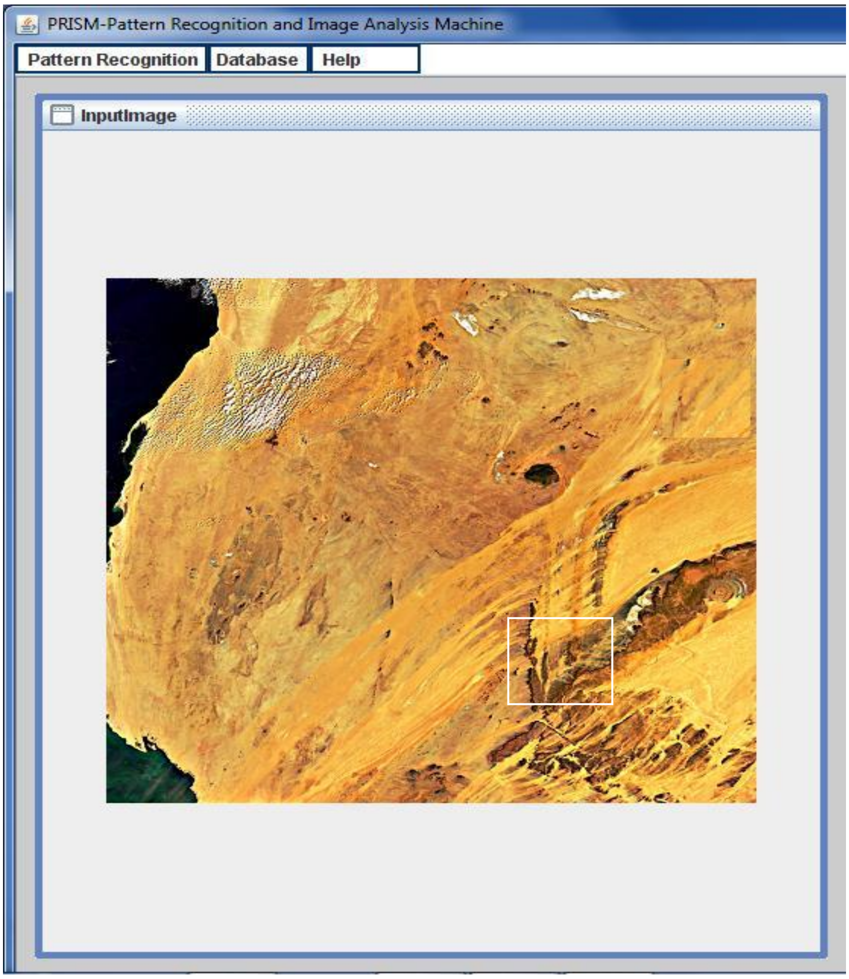
Selection of an object in the input image



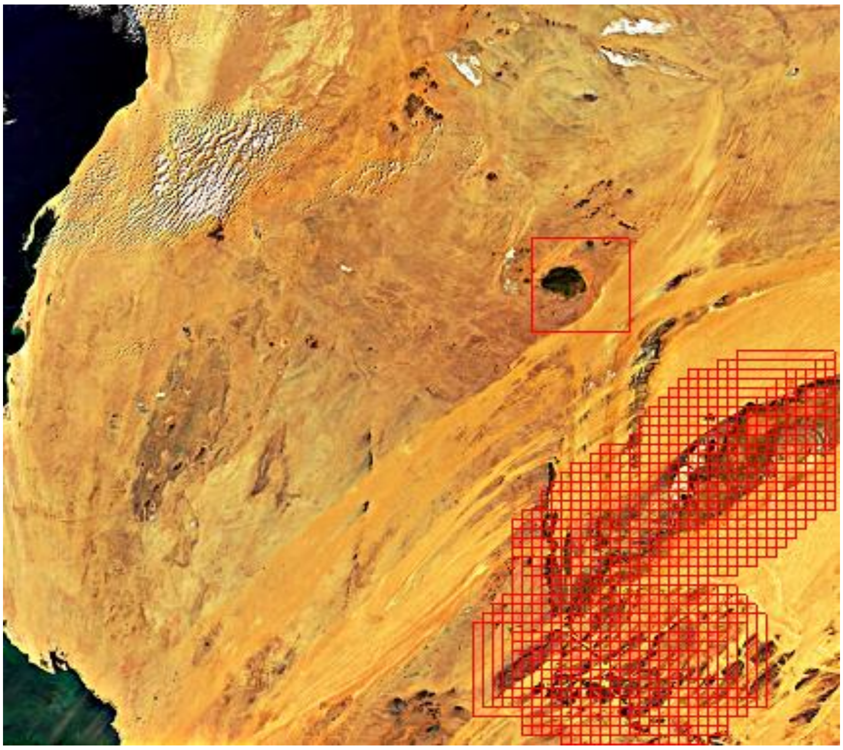
Objects Highlighted



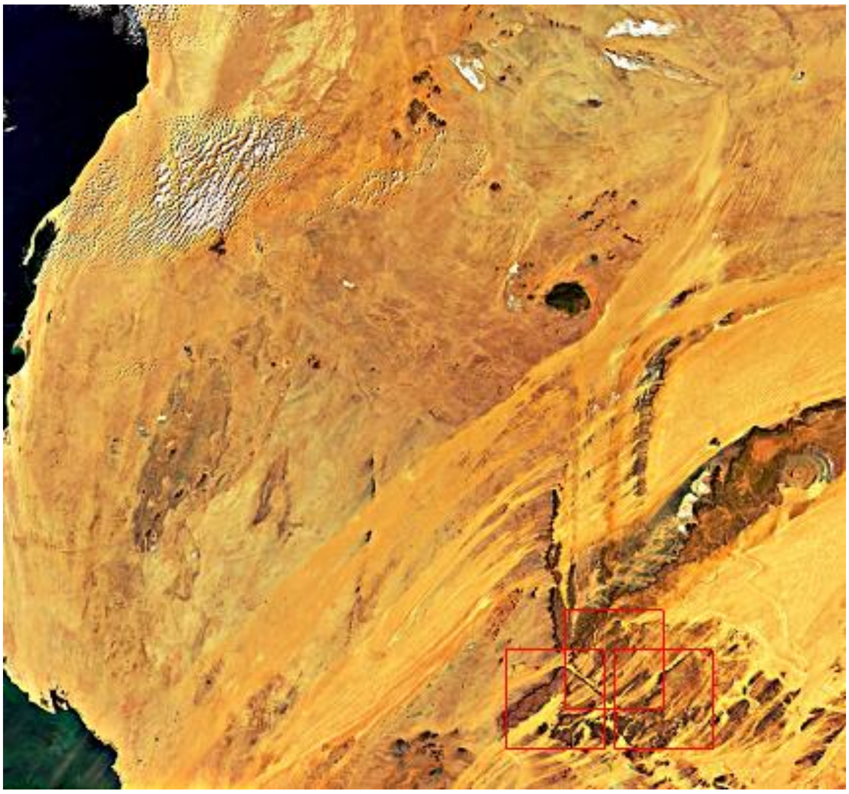
Input for identifying similar texture in the input image of Marituania desert region



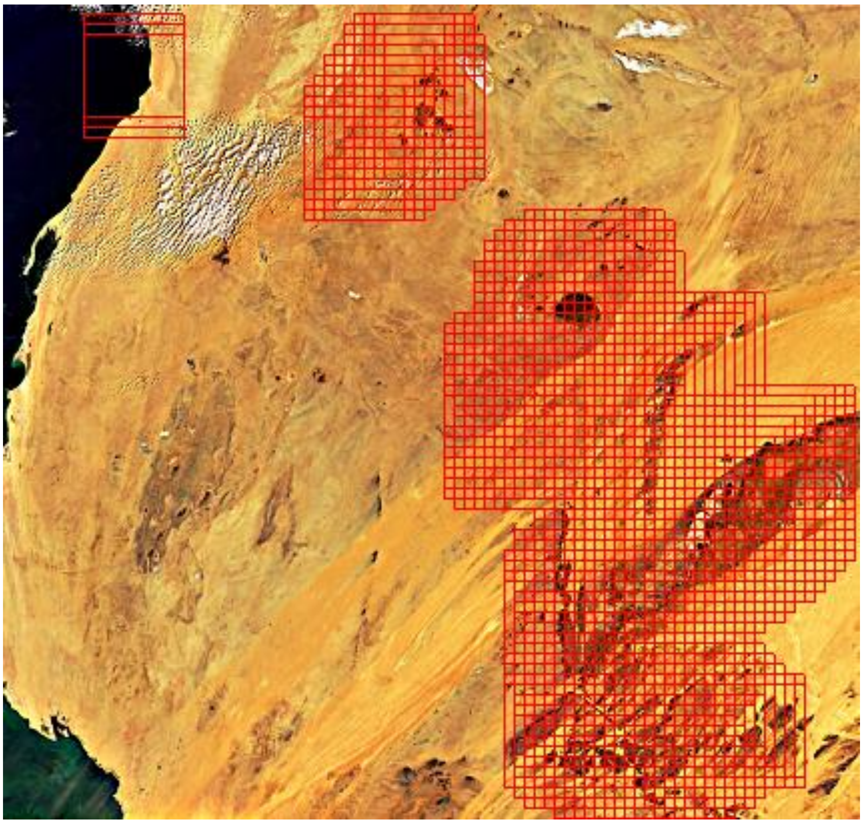
The texture identified by the glcm approach



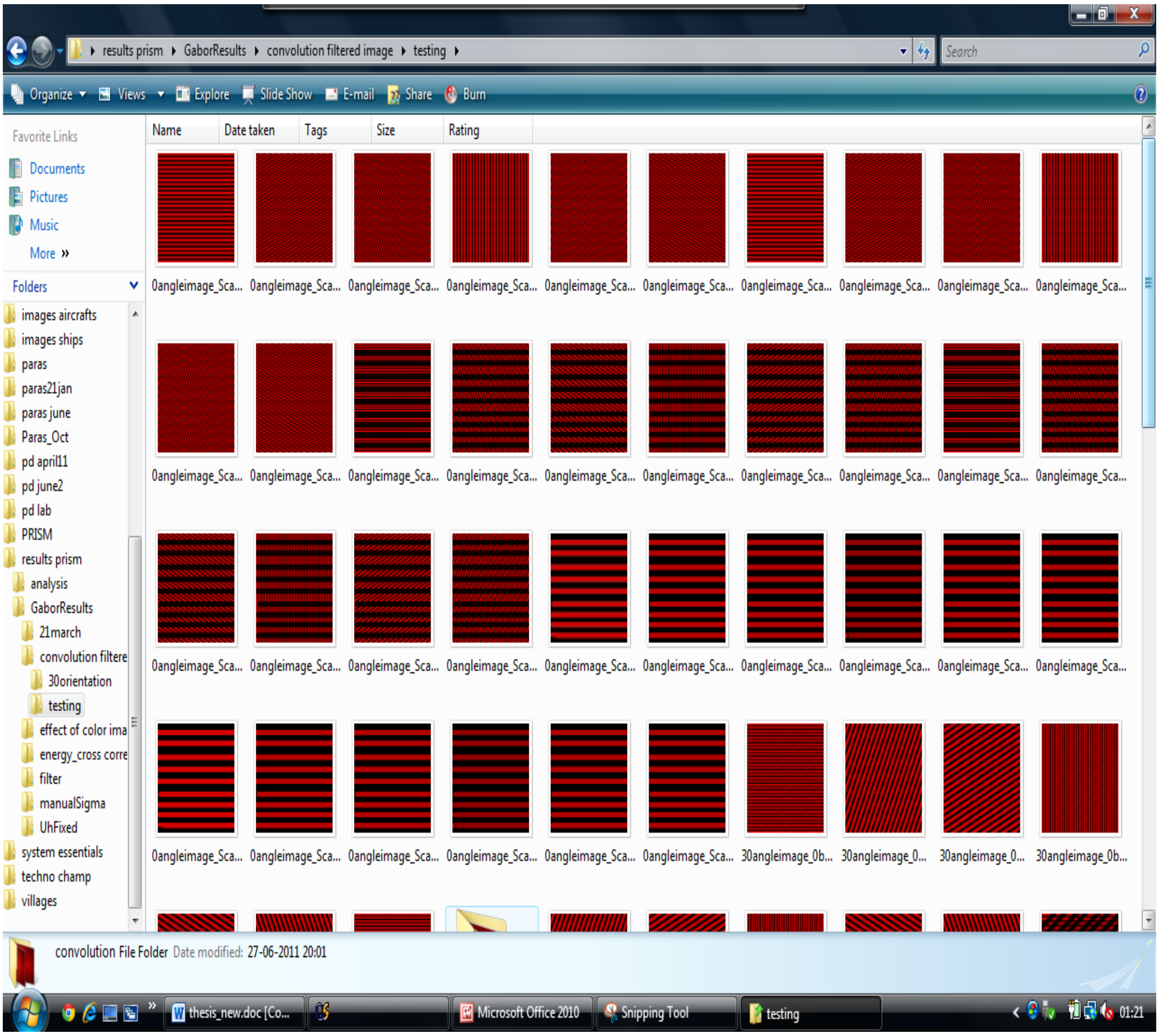
Texture features identified by the gabor filter approach



The texture feature identified by the moment invariant technique



Gabor Filters Generated



DATABASE GLIMPSE

Edit Data - PostgreSQL 8.4 (localhost:54

File Edit View Tools Help

	databasefeimagename [PK] character varying(50)
1	abc1.bmp
2	bgy.bmp
3	cheetah.bmp
4	csd.bmp
5	dij1.bmp
6	eye_1610.bmp
7	Hydrangeas.bmp
8	tiger3.bmp
*	

Edit Data - PostgreSQL 8.4 (localhost:5432) - PRISM - databasetargeti

File Edit View Tools Help

No limit

	databasetargetimagename [PK] character varying(50)	description character varying(100)
1	animal.bmp	
2	cheetah.bmp	
3	CLIP37.bmp	
4	Hydrangeas.bmp	
5	imbmp.bmp	
6	LION.bmp	
7	lion1.bmp	
8	lion2.bmp	
9	tiger.bmp	
10	tiger1.bmp	
11	tiger2.BMP	
12	tiger3.bmp	
13	Tulips.bmp	
*		

Tables indicating the names of images in the database for Object based and Target Based Applications

Edit Data - PostgreSQL 8.4 (localhost:5432) - PRISM - bgycolormoments

File Edit View Tools Help

No limit

	cord_x integer	cord_y integer	r_mean numeric	g_mean numeric	b_mean numeric	r_sd numeric	g_sd numeric	b_sd numeric	r_skew numeric	g_skew numeric	b_skew numeric
1	0	0	88.724	73.024	74.1948	12.9814569290	22.1651759298	22.4650151480	7.87208065965	-24.447077214	-27.026274637
2	0	5	89.3888	73.8168	73.1004	12.4632593874	21.6323836356	23.7999137477	7.65976544129	-23.772987205	-28.288741487
3	0	10	89.5416	73.9276	72.4752	12.3270219209	21.6167748343	24.6696938351	7.77167396905	-23.701530206	-29.187580794
4	0	15	89.5984	73.934	71.5972	12.1606709288	21.5775965149	25.5578198131	8.00542187253	-23.625346699	-30.077126228
5	0	20	90.07	74.3892	71.0684	11.9975455823	21.4578193673	26.1984527024	7.61108508192	-23.550284009	-30.636724953
6	0	25	90.92	75.262	71.478	11.5545661969	21.0134985188	25.6541941989	6.26875290933	-23.233801270	-29.774533427
7	0	30	90.9068	75.0564	71.112	10.9963863955	20.5854997316	25.3777848331	-2.9361861013	-23.025384494	-29.278606003
8	0	35	91.3268	75.4312	70.1756	11.0363762965	20.6252312937	26.2309974282	-6.2286498557	-23.327452251	-30.030624178
9	0	40	91.7768	75.88	70.2564	11.0771016859	20.6189488151	26.5257436314	-7.2909972189	-23.473052062	-30.201645562
10	0	45	92.1392	76.2724	71.51	10.7382970418	20.3309798209	26.0143228364	-7.9287028340	-23.263414493	-29.458515137
11	0	50	92.5312	76.6696	72.9056	10.6909787465	20.3123703786	25.9571351269	-8.2686407128	-23.300914671	-29.329286631
12	0	55	93.4152	77.8572	74.514	10.8932827448	20.3978080234	25.7775166654	-8.4898544144	-23.405481038	-29.168458124
13	0	60	94.2748	79.0644	76.1816	10.8874645790	20.3038843771	25.3605147361	-8.8343174420	-23.311603282	-28.810562367
14	0	65	95.4728	80.6796	78.1528	10.8143451100	20.0854103786	25.0058683544	-8.7987586703	-23.024550515	-28.550322907
15	0	70	96.0996	81.5252	79.532	10.6400413457	19.8568502114	24.5257894013	-8.5784890534	-22.751533732	-28.266897334
16	0	75	95.4732	80.7344	80.5516	11.2296964233	20.5031529789	23.3240794888	-9.0809364301	-23.610529348	-27.197413207
17	0	80	94.8476	80.01	81.2412	12.1644553614	21.3497043014	22.3424516004	-9.8969277933	-24.675085293	-26.267546584
18	0	85	94.4608	79.598	82.2872	12.6173080869	21.7752524632	20.8572206470	-10.023482930	-25.137825814	-24.573885433
19	0	90	94.2528	79.5388	83.156	12.9023134421	22.1320195770	20.0421813742	-9.8110370252	-25.427184709	-23.905547285
20	0	95	94.1248	79.602	83.65	13.4678441095	22.7753400817	19.5103904379	-9.5859127635	-25.923353700	-23.735992043
21	0	100	94.1096	79.876	84.33	13.7047140736	23.0679343019	18.7655449204	-9.6057006993	-26.191747200	-23.222187696
22	0	105	94.056	80.0864	85.4292	13.6691793462	23.0506368502	18.4027896146	-9.4252947125	-26.110994766	-22.971455668
23	0	110	93.7488	79.9956	86.3424	13.4878945191	22.7899559209	17.7174017056	-8.9196014183	-25.797908342	-22.194189345
24	0	115	93.0568	79.3816	87.1076	13.0396922417	22.3065881855	16.9732319515	-8.4338969688	-25.287246988	-21.105722968
25	0	120	92.798	79.3696	88.2076	12.8937968031	22.0546168046	16.2634644033	-7.6669968814	-24.873891360	-19.781143043
26	0	125	93.9108	81.2108	90.0344	12.5994461529	21.3411987329	15.6980410752	-7.4226675835	-24.059207858	-19.239278762
27	0	130	95.2848	83.1896	92.1428	12.1001689641	20.4593918501	14.8290853446	-2.5800894362	-22.743000399	-18.542701673
28	0	135	96.304	84.7668	93.782	11.4157252945	19.4312789491	14.0827184875	5.98466802669	-21.347470273	-18.074496824
29	0	140	97.048	85.9064	95.2256	11.0456550733	18.6155228129	13.3261789872	7.36888661771	-20.125491387	-16.978780501
30	0	145	97.9412	87.1572	96.4384	10.4125377579	17.5591555651	12.3530001732	8.52505116054	-18.409722333	-15.568031957
31	0	150	98.6016	88.1356	97.6228	10.1548253278	16.9378619855	11.4859640257	9.47772298131	-16.884560048	-13.806585633
32	0	155	98.7108	88.3476	97.7992	10.4914423870	16.9675539922	11.5343094253	9.89412580463	-16.551696536	-12.932278446
33	0	160	99.7336	89.746	97.9444	11.2451336603	17.4851261865	11.7775441107	9.99590288338	-16.959364002	-13.619296233
34	0	165	101.3984	91.9896	98.0896	11.7041051533	17.6873686364	11.7971576780	9.40063299334	-17.385845971	-14.237543934
35	0	170	103.1648	94.3472	98.1148	11.5365870585	17.2463770665	12.0776869043	8.17198062920	-17.378332074	-15.057451675

4900 rows.

The Color moments of different templates formed in an image

Database Snapshot of the Images Used

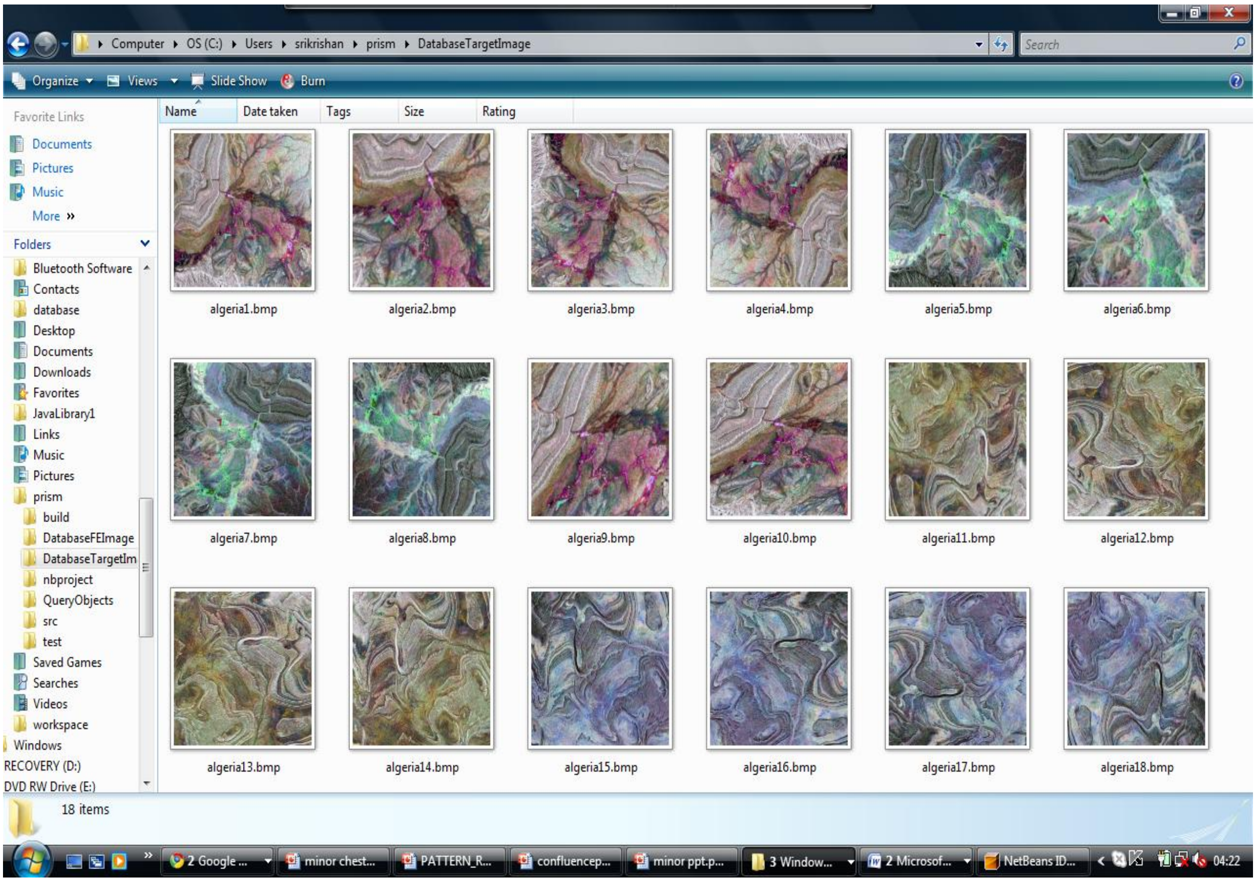


IMAGE UNDERSTANDING RESULTS

Image used as input for image understanding

Output obtained

The image has following characteristics:

Aircrafts

Metalled Road

Conclusion: Runway

Input image used:

Output obtained:

The image has following characteristics:

Aircrafts

Metalled Road

Conclusion: Runway

Input image used:

Output obtained:

The image has following characteristics:

Aircrafts

Metalled Road

Conclusion: Storage Unit: Terrorist Airbase camp

Input image used:

Output obtained:

The image has following characteristics:

Aircrafts

Metalled Road

Conclusion: Storage Unit: Hangar

APPENDIX 3:PUBLICATION

Conference Name: International IT Summit, ‘Confluence – The Next Generation Information Technology’

Paper Title: Feature Extraction Methods for Content Based Image Retrieval

Authors: Chesta Agarwal

Location: Delhi, INDIA

Remarks: Won the Best Research Paper Presentation at the Conference