MAJOR PROJECT REPORT

ON

OBJECT RECOGNITION BY FUZZY CLASSIFICATION OF GABOR WAVELET FEATURES

Submitted in partial fulfilment of the requirements

For the award of the degree of

Master of Technology
In
Information Systems

Submitted By:

SEEMA CHANDNA

Roll No. 13/ISY/2K10

Under the Guidance of

Ms. Seba Susan

(Assistant Professor, IT Department)

Department of Information Technology

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi-110042

2010-2012

CERTIFICATE

This is to certify that the work contained in the thesis titled *“**Object Recognition by Fuzzy Classification of Gabor Wavelet Features**”* is an original piece of work which has been carried out by **Seema Chandna** (13/ISY/2K10) for the award of the degree of ***Master of Technology in Information System*** (Department of Information Technology) under my supervision. This work has not been submitted elsewhere for any degree.

Ms. Seba Susan

(Project Guide)

Assistant Professor

Department of Information Technology

Delhi Technological University, Delhi

June 2012

ABSTRACT

Gabor features, a well-researched topic, widely used in image processing applications such as object and face recognition, also pattern recognition applications such as fingerprint recognition, character recognition, and texture segmentation etc. Object recognition in the field of computer vision focuses on the task of identifying or locating a set of objects in an image or video sequence. It is a problem of matching models from a database with representations of those models extracted from the image luminance data. A novel system for Object Recognition based on fuzzy classification has been proposed in this research. Here, we deal with Otsu thresholding to binarize the object image; features extracted using Gabor Wavelet and then applying our proposed fuzzy classifier for object recognition. The fuzzy classifier used, is based on the Generalised Gaussian membership function. We use well known Caltech dataset for our work. Experimental results illustrate the efficiency of the proposed method. The performance of the proposed classifier has been compared against some of the other commonly known classifiers for object recognition. In our work, we have used MLP-Neural Network classifier, SVM classifier, Naive bayes, Nearest Neighbour classifier and proposed fuzzy classifier in object recognition for comparison. Here, we have also compared the result of paper [47] having its own classifier and with MLP-Neural Network classifier. With our proposed approach we are able to obtain 74% Recognition Rate or we can say 74% of test objects were correctly classified. It can be seen that our proposed system gives highest performance in comparison to other systems. We also showed that our new approach has considerable advantages and is substantially superior to the existing traditional methods especially in the field of robustness and accuracy.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Ms. Seba Susan for giving me an opportunity to work with her and for her advice, encouragement and constant support. I wish to thank her for extending me the greatest freedom in deciding the direction and scope of my research. It has been both a privilege as well as a rewarding experience working with her.

I would also like to thank all my friends and colleagues here at Delhi Technological University for their valuable comments and suggestions, many of which have been vital to this work. Furthermore, I want to thank the faculty of Department of Information Technology and the staff for providing me the means to complete my degree.

I would like to thank my parents for their love, sacrifice, understanding, encouragement and support. I am forever indebted to them for all that they have given me. Finally, special thanks to my brother for his love, affection and help.

SEEMA CHANDNA

Roll No. 13/ISY/2K10

M.Tech (Information Systems)

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 What is Computer Vision?

Humans are capable of recognizing the world around them based on the visual clues which they perceive from the environment. Computer vision is the science which aims a machine or computer to achieve similar abilities of human vision by electronically perceiving and understanding an image. It combines knowledge from computer science, electrical engineering, mathematics, physiology, biology, and cognitive science in order to grasp and simulate the operations of the human vision system. During evolution, the input sensor for the biological vision, the eye, was "invented" at least 38 times according to the zoologists Ernst Mayr and Luitfried von Salvini-Plawen [80]. It is a logical step in the evolution of computers and robots to also develop in this direction.

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding information from a single image or a sequence of images. In general, Computer vision (image understanding) is a discipline that studies how to reconstruct, interpret and understand high-dimensional data from the real world and produce numerical or symbolic results in the forms of decisions. The goal of computer vision is to process images acquired with cameras and generates a representation of objects in the world. Computer vision is difficult when we acquire or analyze noisy image data or data with uncertainties. It overlaps with the fields like image processing, pattern recognition, and photogrammetry. Image processing deals with image manipulation to enhance image quality, to restore an image or to compress/decompress an image. Most computer vision algorithms usually assume a significant amount of image processing has taken place to improve image quality. Pattern recognition studies various techniques (such as statistical techniques, neural network, support vector machine, etc.) to recognize/classify different patterns. Pattern recognition techniques are widely used in computer vision. Photogrammetry is concerned with obtaining accurate and reliable measurements from images. It focuses on accurate measurements. Camera calibration and 3D reconstruction are two areas of interest to both computer vision and photogrammetry researchers. Computer vision finds application in many areas such as:

- Medical image analysis where information is extracted from image data (in the form of microscopy images, X-ray images, ultrasonic images) for the purpose of making a medical diagnosis of a patient.
- In industry also called machine vision where information is extracted for the purpose of supporting a manufacturing process or for industrial quality inspection.

- Biometrics where information is extracted for identification or verification using face recognition, voice recognition or fingerprint recognition.
- Military applications where information is extracted for detection of enemy soldiers or vehicles and missile guidance.
- Autonomous vehicles use computer vision for navigation, for producing a map of its environment and for detecting obstacles.
- Other application fields includes agriculture, augmented reality, character recognition, forensics, gesture analysis, geosciences, image restoration, pollution monitoring, process control, remote sensing, Robotics, security and surveillance and transport.

1.2 Pattern Recognition

"The assignment of a physical object or event to one of several pre-specified categories" -- Duda & Hart

Pattern recognition is an important field of computer science whose objective is to recognize patterns, which can be visual or audio patterns. It can also be said as the science of making inferences based on data. The term pattern recognition is comprised of two words pattern and recognition.

- A **pattern** is an entity, process or event that can be given a name or category, based on features derived to emphasize commonalities. e.g., fingerprint image, handwritten word, human face, speech signal, DNA sequence. A pattern class (or category) is a set of patterns sharing common attributes and usually originating from the same source. In practice, features are often extracted from sensory signals, such as images or audio, and this step distinguishes pattern recognition from fundamental statistical classification whose starting point follows data acquisition.
- **Recognition (or classification)** implies an act of assigning input values to prescribed classes. A classifier is a machine which performs classification. In a broad sense, recognition associates classification with a label. Using the Figure 1.1, that would say that those samples which are falling into the upper right region are recognized as one class (say A) and those in the lower left are recognized as other class (say B). Strictly speaking, pattern recognition doesn't go that far. Pattern recognition says only that the upper right samples are classified as having heavier weight and longer height. Determining that this category as 'A' is one step above pattern recognition. In this sense, pattern classification more correctly describes this field.

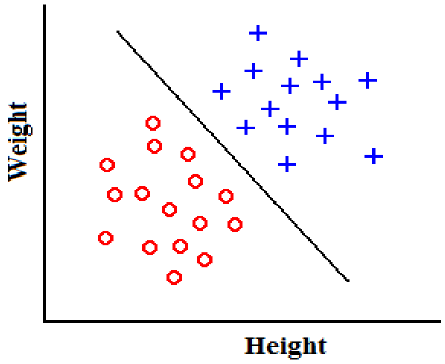


Figure 1.1: Recognition method.

So pattern recognition is the study of how machines can observe the environment; how they learn to distinguish patterns of interest; and how they make sound and reasonable decisions about the categories of the patterns. Pattern recognition is an active research field over the past few years due to its scientific interest and potential applications such as

- Biometrics (voice recognition, automatic recognition of images of human faces, fingerprints recognition),
- Diagnostic systems (medical diagnosis: X-Ray, EKG analysis, machine diagnostics, waster detection),
- Military applications (automated target recognition, Image segmentation and analysis (recognition from aerial or satelite photographs)) and,
- Optical character recognition (OCR) (handwriting recognition, classification of text into several categories (e.g. spam/non-spam email messages), the automatic recognition of handwritten postal codes on postal envelopes).

Pattern Recognition System: Pattern recognition can be defined narrowly as dealing with feature extraction and classification as shown in Figure 1.2. However, tools and methods of the field can be applied broadly. Much of the effort at this time seems to be concentrated in just a couple of application areas. While this depth of investigation is important, it is my belief that researchers achieve innovation as well by choosing novel and neglected problems. By pursuing breadth as well as depth we can have impact in two ways: by advancing the task at hand and by influencing others in our field to explore a variety of interesting problems. ~Larry O'Gorman

- **Data acquisition and preprocessing:** Collects the data from several sensors, measures physical variables. Important issues which must be take care are bandwidth, resolution, sensitivity, distortion, SNR, latency, etc. While preprocessing noise is removed from data in order to assure it satisfies certain assumptions implied by method, patterns of interest are isolated from the background.

- **Feature extraction:** Extracts a set of salient or discriminatory features after processing the acquired data.
- **Learning algorithm:** Learning a mapping between features and pattern groups and categories with pattern recognition from training examples.
- **Classification:** Using features and learned models it assigns a pattern to a category.

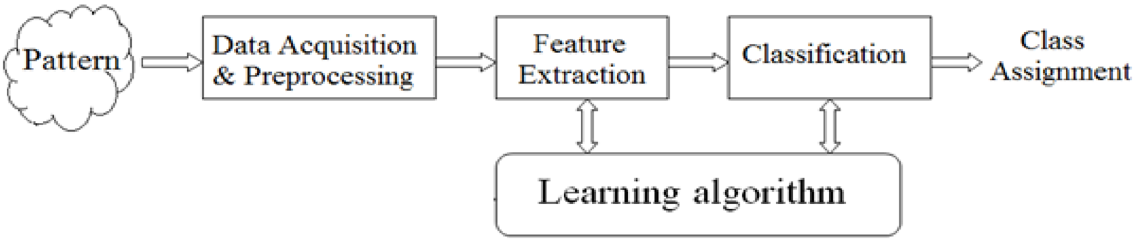


Figure 1.2: Pattern Recognition System.

We are often influenced by the knowledge of how patterns are modelled and recognized in nature when we develop pattern recognition algorithms. Research on machine perception also helps us gain deeper understanding and appreciation for pattern recognition systems in nature. The main aim in research of pattern recognition is to measure the performance of the classifier, i.e. by the error rate, which is defined as the ratio of misclassifications to the total number of patterns seen in the evaluation. In image recognition the basic problem is to identify the objects present in the given image. Humans perform with intelligence, but it is very difficult to teach a digital computer. For a computer, a digital image consists of an array of pixel values, which has no associated meaning in itself. An object class can be defined by its functionality, such as chairs, or by its shape, such as cylinders, or by its intrinsic nature, such as mammals. As a result, the possible features that are relevant in object recognition are many, and vary in importance for each class and defining each class is nearly impossible. Dealing with image object recognition, in almost every case one is interested in designing classifiers that can tolerate certain transformations of the input patterns, that is, invariant recognition of the image content must be achieved [81].

1.3 Image Features

Image processing belongs to the field of signal processing in which input and output signals are both images. In computer vision society, the concept of feature is used to denote a piece of information which specifies quantifiable property of an image, and is computed such that it quantifies some significant characteristics of that image and is used for solving the computational task related to a certain application. More specifically, features can refer to specific structures in the image such as simple structures like points or edges or complex structures like objects. In addition, the computation efficiency is also an important factor for image feature extraction and description. Image features can be classified as follows:

- **General features:** Application independent features such as color, texture, and shape. According to the abstraction level, they can be further divided into:
  - **Pixel-level features:** The most direct approach to query for images is to compare the images directly. Features calculated at each pixel, e.g. color, location. That is, the pixel values of the image itself or the pixel values of a scaled version are compared directly to the corresponding values of other images. For many applications this approach is not feasible as it is not clear which pixels from the one image correspond to which pixels in the other image. In optical character recognition this method is suitable when the symbols are already segmented since a letter to be recognized will probably be similar to another observation of the same letter when the letters are of equal size and contained at the same position in the image. In addition to taking the pixel values themselves several extensions are possible [83].
  - **Local Approach:** Local approach calculates features over the results of subdivision of the image band on image segmentation or edge detection or we can say it extracts features by 1st dividing the object into smaller regions. In general it represents image patches and features are computed at multiple points in the image and are consequently more robust to occlusion and clutter. However, they may require specialized classification algorithms to handle cases if there is variable number of feature vectors per image and making comparison of images complicated. These extracted features are used for recognizing the objects. Examples for local features of an object are e.g., the color, gradient or small region. For object recognition tasks the local feature should be invariant to illumination, noise, scale and viewpoint, but, in general, this cannot be reached completely due to the simplicity of the features itself. Thus, several features of a single point or distinguished region in various forms are combined and a more complex description of the image is obtained which is referred as descriptor. One advantage of using local features is that they are able to recognize the object despite of significant clutter and occlusion. They also do not require a segmentation of the object from the background, unlike many texture features, or

representations of the object’s boundary (shape features). The techniques which are using local approach are SIFT (Scale Invariant Feature Transform) features proposed by Lowe [87] uses local maxima of the difference-of-Gaussians function as interest points and histograms of gradient orientations computed around the points as the descriptors; Hausdorff Average, a standard technique for comparing point sets of different sizes, and apply it to comparing images represented with local features; Probabilistic method evaluates the average log likelihood of feature points of a particular image under a non-parametric density estimate for the class.

- **Global Approach:** Global approach calculates features from the input data i.e. entire image or sub-area of an image. In general it describes an image as whole and has the ability to generalize an entire object with a single vector. Features extracted are general properties of the object such as color histograms [88], contour representations, texture features and shape descriptors [19] [90]. Such features represents image in a high dimensional feature space. So any standard classifier can be used. Global approaches are difficult due to segmentation, non-uniform background, and clutter and occluded objects. These approaches varies from simple statistical measures (e.g., mean values or histograms of features) to more sophisticated dimensionality reduction techniques, such as principle component analysis (PCA) [?57], independent component analysis (ICA) [?53], or non negative matrix factorization (NMF). The main idea of all of these methods is to project the original data onto a subspace, that represents the data optimally according to some predefined criterion: minimized variance (PCA), independency of the data (ICA), or non-negative, i.e., additive, components (NMF). The global texture features can be used such as local binary patterns (LBP), which are gray-scale and rotation invariant texture operators, and shape index which is computed using the isophote and the flow line curvatures of the intensity surface.

Global texture features and local features provide different information about the image because the support over which texture is computed varies. Despite the robustness advantages of local features, global features are still useful in applications where a rough segmentation of the object of interest is available. An image often does contain a single object, but sometimes several organisms or particles are present. It should be noted that in such cases a vision system for object recognition may not be devoted to either local or global approach only but may select part based approaches i.e. simple global bimodal segmentation, which is effective for separating the object from the background. The basic advantage is that the background and occluding objects can be more easily discarded by simply looking only at the features on the object. Other advantage is that modeling the appearance of a subset of the object parts is much easier task than modeling an object class as a whole [84-86].

- **Domain-specific features:** Application dependent features such as human faces, fingerprints, and conceptual features. These features are often a synthesis of low-level features for a specific domain [88].

On the other hand, all features can be coarsely classified into low-level features and high-level features. Low-level features are the features which can be extracted directly from the original images without any information about the shape. A widely used approach is the so called edge detection, which is adopted in order to identify points in a digital image at which the image brightness changes abrasively, also edge detection highlights image contrast. The boundary of features within an image can be discovered detecting contrast as the difference in intensity. Whereas high-level feature extraction must be based on low-level features and is used to find shapes in computer images. To better understand this approach let us suppose that the image to be analyzed is represented by a human face. If we want to automatically recognize the face, we can extract the component features, for example the eyes, mouth and the nose. To detect them we can exploit their shape information: for instance, we know that the white part of the eye is ellipsoidal and so on. Shape extraction includes finding the position, the orientation and the size. In many applications the analysis can be helped by the way the shapes are placed. In face analysis we imagine to find eyes above and the mouth below the nose and so on.

1.3.1 Color

Color plays a very important role in current state-of-the-art content-based image retrieval from image databases. The color feature is one of the most widely used visual features in image retrieval. Images characterized by color features have many advantages:

- ***Robustness -*** The color histogram is invariant to rotation of the image on the view axis, and changes in small steps when rotated otherwise or scaled. It is also insensitive to changes in image and histogram resolution and occlusion.

- ***Effectiveness -*** There is high percentage of relevance between the query image and the extracted matching images.

- ***Implementation simplicity*** **-** The construction of the color histogram is a straightforward process, including scanning the image, assigning color values to the resolution of the histogram, and building the histogram using color components as indices.
- ***Computational simplicity*** **-** The histogram computation has O(X,Y) complexity for images of size X × Y. The complexity for a single image match is linear, O(n), where n represents the number of different colors, or resolution of the histogram.
- ***Low storage requirements*** **-** The color histogram size is significantly smaller than the image itself, assuming color quantization.

Typically, the color of an image is represented through some color model. A color model is specified in terms of 3-D coordinate system and a subspace within that system where each color is represented by a single point. The more commonly used color models are RGB (red, green, blue), HSV (hue, saturation, value) and Y, C_b, C_r (luminance and chrominance). Thus the color content is characterized by 3-channels from some color model. One representation of color content of the image is by using color histogram. Statistically, it denotes the joint probability of the intensities of the three color channels. Color is perceived by humans as a combination of three color stimuli: Red, Green, and Blue, which forms a color space. By varying their combinations, other colors can be obtained. The representation of the HSV space is derived from the RGB space cube, with the main diagonal of the RGB model, as the vertical axis in HSV. As saturation varies from 0.0 to 1.0, the colors vary from unsaturated (gray) to saturated (no white component). Hue ranges from 0 to 360 degrees, with variation beginning with red, going through yellow, green, cyan, blue and magenta and back to red. So these color spaces corresponds to the RGB model from which they can be derived through linear or non-linear transformations. For a three-channel image, there are three of such histograms. The histograms are normally divided into bins in an effort to coarsely represent the content and reduce dimensionality of subsequent matching phase. A feature vector is then formed by concatenating the three channel histograms into one vector. For image retrieval, histogram of query image is then matched against histogram of all images in the database using some similarity metric.

- **Color Histograms:** In image retrieval systems color histogram is the most commonly used feature. The main reason is that it is independent of image size and orientation. Also it is one of the most straight-forward features utilized by humans for visual recognition and discrimination. Statistically, it denotes the joint probability of the intensities of the three color channels. Once the image is segmented, from each region the color histogram is extracted. The major statistical data that are extracted are histogram mean, standard deviation, and median for each color channel i.e. Red, Green, and Blue. So totally 3 × 3 = 9 features per segment are obtained. All the segments need not be considered, but only segments that are dominant may be considered, because this would speed up the calculation and may not significantly affect the end result. Color descriptors of images can be global or local and consist of a number of histogram descriptors and color descriptors represented by color moments, color coherence vectors or color correlograms [88]. Color histogram describes the distribution of colors within a whole or within an interest region of image. The histogram is invariant to rotation, translation and scaling of an object but the histogram does not contain semantic information, and two images with similar color histograms can possess different contents. A color histogram H for a given image is defined as a vector $H = \{ h[1], h[2], . . . h[i], . . . , h[N] \}$ where i represents a color in the color histogram, h[i] is the number of pixels in color i in that image, and N is the number of bins in the color histogram, i.e., the number of colors in the adopted color model. In order to compare images of different sizes, color histograms should be normalized. The normalized color histogram H is defined for $h'[i] =h[i]XY$ where XY is the total number of pixels in an image (the remaining variables are defined as before). A problem with histograms is the discontinuity. That is, slightly changing the image might change the bin assignments and thus the resulting histogram completely. To overcome the problem fuzzy histograms can be used. The goal of fuzzy histograms is to remove the discontinuous bin assignment of the traditional histogram. Objects were also represented by a color histogram [89]. Objects are identified by matching histograms of image regions to histograms of a model image. While the technique is robust to object orientation, scaling, and occlusion, it is very sensitive to lighting conditions, and it is not suitable for recognition of objects that cannot be identified by color alone. Many other approaches were also proposed to exploit illumination invariants [89].

1.3.2 Texture

Texture is another important property of images. Texture can be defined as the visual patterns that have properties of homogeneity and do not result from the presence of only a single color or intensity. It is an innate property of virtually all surfaces, including cloud, trees, bricks, hair, fabric, etc. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. In computer vision society, texture is a kind of very interesting and challenging property of images. It has been exploited extensively for various kinds of special purposes, such as for surface description, image matching, and 3D-reconstruction etc. Texture is a powerful regional descriptor and is ideally suited for the retrieval process such as medical image retrievals. Texture, on its own does not have the capability of finding similar images, but it can be used to classify textured images from non-textured ones and then be combined with another visual attribute like color to make the retrieval more effective. Texture has been one of the most important characteristic which has been used to classify and recognize objects and have been used in finding similarities between images in multimedia databases. A texture can be characterized by a set of values called energy, entropy, contrast, and homogeneity. Textures can be categorized as following four types [90]:

1. ***Surface Texture* -** Created by the regular repetition of an element or pattern, called *surface texel*, on a surface.

2. ***Image Texture* -** The image of a surface texture, itself a repetition of i*mage texels*, the shape of which is distorted by the projection across the image.

3. ***Deterministic Texture* -** Created by the repetition of a fixed geometric shape such as a circle, a square, a decorative motif. For example, patterned wallpaper, bricks wall, and decorative tiles. This kind of texture can be represented naturally by the shape parameters of the specific shape at hand.

4. ***Statistic Texture* -** Created by changing patterns with fixed statistical properties. For example pebbles, gravel, wood, or lawns. This kind of texture can be represented typically in terms of spatial frequency properties. For instance, represented by the power spectrum computed over image regions.

Texture segmentation is often obtained by adopting the independent sub-processes of texture feature extraction, feature selection or reduction if the number of features is too large, followed by a segmentation algorithm. The main purpose of texture feature extraction is to map differences in spatial structures, either stochastic or geometric, into differences in gray value. Segmentation methods then analyze the feature space in order to extract homogeneous regions. Texture segmentation methods may be categorized as:

- **Feature-based:** Some characteristic or characteristics of textures are chosen and regions in which these characteristics are relatively constant (or the boundaries between the regions) are sought. **Methods using unique features** include *operator-based features* (Laplacian operator); *statistic-based features* (it includes methods like Fourier power spectra, shift-invariant principal component analysis (SPCA) or Spatial gray level dependence method. In contrast, texture features can be coarseness, contrast, directionality, line-likeness, regularity, and roughness.); *transform domain features* (initially reducing the dimension and then performing the segmentation). **Methods using unique segmentation techniques** include region-based methods; boundary-based methods; hybrid methods.

- **Model-based:** It models underlying processes for textures and segments using certain parameters of these processes. In this paradigm the simplifying assumption is made that the environment consists of manufactured objects whose geometry is known a priori. This method include ***fractal models*** (it has found a high degree of correlation between fractal dimension and human estimates of roughness and can be considered over a limited range of scales); ***stochastic models and Decision-theoretic techniques*** include Simultaneous Autoregressive (SAR) random field models, Markov Random Field (MRF), Gibbs distribution (in fact equivalent to MRF), Gaussian Markov Random Field (GMRF); and ***multi-resolution filtering techniques*** such as Gabor and wavelet transform.
- **Structure-based:** Seek to partition images under the assumption that the textures in the image have detectable primitive elements, arranged according to placement rules.

1.3.3 Shape

Shape based image retrieval is the measuring of similarity between shapes represented by their features. Shape is an important visual feature and it is one of the primitive features for image content description. Shape content description is difficult to define because measuring the similarity between shapes is difficult. Therefore, two steps are essential in shape based image retrieval, they are: feature extraction and similarity measurement between the extracted features. Shape descriptors can be divided into two main categories: region-based and contour-based methods. *Region-based methods* use the whole area of an object for shape description, while *contour-based methods* use only the information present in the contour of an object. The shape descriptors described are [90]:

• ***Shape descriptors*** - features calculated from objects contour: circularity, aspect ratio, discontinuity angle irregularity, length irregularity, complexity, right-angleness, sharpness, directedness. Those are translation, rotation (except angle), and scale invariant shape descriptors. It is possible to extract image contours from the detected edges. From the object contour the shape information is derived. We extract and store a set of shape features from the contour image and for each individual contour.

• ***Region-based shape descriptor -*** utilizes a set of the geometrical moments, central moments and normalized central moments, moment invariants, Zernike moments (which are based on the theory of orthogonal polynomials) calculated at the center of the image.

1.3.4 Latest Image Features

Feature extraction is the first and crucial step for obtaining the 3D information and image understanding. Image feature detection and description are the basis for image and video analysis. Robust and efficient image features can significantly improve the performance of image/video retrieval, recognition, tracking, biometrics, matching, registration, reconstruction, etc. Image features can be classified as local, dense, dense local, semi-local and global. In many cases, image features are expected to be stable, repeatable, illumination invariant, discriminative and representative. In recent years, many powerful image features are developed such as SIFT, SURF, HOG, LBP, GLOH, Daisy, GIST (Spatial Envelope), RCM (Region Covariance Matrix). Some of these latest explained here which were used for object recognition purpose are:

1.3.4.1 SIFT descriptor: SIFT features are well-known descriptors used widely through computer vision and object recognition tasks. SIFT is an algorithm for extraction and description of local features. The scale-invariant feature transform (SIFT) descriptor, proposed by David Lowe [87], is arguably one of the most widely used feature representation schemes for vision applications such as object retrieval, and object category discovery. These features are useful because they provide highly descriptive texture-based features which are robust to most changes in scale and rotation. Because it would be too computationally intensive and unnecessarily repetitive to calculate a feature for every pixel location, we calculate SIFT features only at interest points in the image such as areas with rapid intensity change called “corners”. There are numerous corner detectors that can be used. The central idea of feature-based object recognition algorithms lies in finding interest points, often occurred at intensity discontinuity, that are invariant to change due to scale, illumination and affine transformation. Each SIFT feature represents a vector of local image measurements in a manner that is invariant to image translation, scaling, and rotation, and partially invariant to changes in illumination and local image deformations. A typical image will produce several thousand overlapping features at a wide range of scales that form a redundant representation of the original image. The local and multi-scale nature of the features makes them insensitive to noise, clutter and occlusion, while the detailed local image properties represented by the features makes them highly selective for matching to large databases of previously viewed features. The detector/descriptor combination is called scale invariant feature transform (SIFT) and consists of a scale invariant region detector called as difference of Gaussian (DoG) detector and a proper descriptor referred to as SIFT-key. Each key point specifies 4 parameters: 2D location, scale and orientation. The SIFT feature locations are efficiently detected by identifying maxima and minima of a difference-of-Gaussian function in scale space and keypoints with low contrast or poorly localized on an edge are removed. Next, a consistent orientation is assigned to each keypoint and its magnitude is computed based on the local image gradient histogram, thereby achieving invariance to image rotation. At each keypoint descriptor, the contribution of local image gradients are sampled and weighted by a Gaussian, and then represented by orientation histograms. So the SIFT descriptor of a keypoint is finally a histogram of the gradient orientation. A feature vector is formed by measuring the local image gradients in a region around each location in coordinates relative to the location, scale and orientation of the feature. The gradient locations are further blurred to reduce sensitivity to small local image deformations, such as result from 3D viewpoint change. For example, the 16 × 16 sample image region and 4 × 4 array of histograms with 8 orientation bins are often used, thereby providing a 128-dimensional feature vector for each keypoint. Objects can be indexed and recognized using the histograms of keypoints in images. SIFT-based methods are expected to perform better for objects with rich texture information as sufficient number of keypoints can be extracted. On the other hand, they also require sophisticated indexing and matching algorithms for effective object recognition [85].

1.3.4.2 PCA-SIFT or Gradient PCA: Instead of gradient histograms on DoG-points, the authors applied Principal Component Analysis (PCA) to the scale-normalized gradient patches obtained by the DoG detector. In principle they follow Lowe’s approach for key-point detection. They extract a 41 × 41 patch at the given scale centered on a key-point, but instead of a histogram they describe the patch of local gradient orientations with a PCA representation of the most significant eigenvectors (that is, the eigenvectors corresponding to the highest eigenvalues). In practice, it was shown, that the first 20 eigenvectors are sufficient for a proper representation of the patch. The necessary eigenspace can be computed off-line. In contrast to SIFT-keys, the dimensionality of the descriptor can be reduced by a factor about 8, which is the main advantage of this approach. Evaluations of matching examples show that PCA-SIFT performs slightly worse than standard SIFT-keys [85].

1.3.4.3 Gradient Location-Orientation Histogram (GLOH): Gradient location-orientation histograms are an extension of SIFT-keys to obtain higher robustness and distinctiveness.

Instead of dividing the patch around the key-points into a 4 × 4 regular grid, Mikolajczyk and Schmid divided the patch into a radial and angular grid [95], in particular 3 radial and 8 angular sub-patches leading to 17 location patches (see Figure 1.3) . The idea is similar to that used for shape context. Gradient orientations of those patches are quantized to 16 bin histograms, which in fact results in a 272 dimensional descriptor. This high dimensional descriptor is reduced by applying PCA and the 128 eigenvectors corresponding to the 128 largest eigenvalues are taken for description [85].

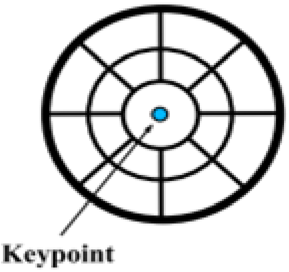


Figure 1.3: GLOH patch scheme.

1.3.4.4 Locally Binary Patterns: Locally binary patterns (LBP) are a very simple texture descriptor approach initially proposed by Ojala et al. [94]. They have performed very well in various applications, including texture classification and segmentation, image retrieval and surface inspection and are based on a very simple binary coding of thresholded intensity values. The original LBP operator labels the pixels of an image by thresholding the 3-by-3 neighbourhood of each pixel with the center pixel value and considering the result as a binary number (as shown in Figure 1.4). The 256-bin histogram of the labels computed over an image can be used as a texture descriptor. Each bin of histogram (LBP code) can be regarded as a micro-texton. Local primitives which are codified by these bins include different types of curved edges, spots, flat areas, etc.

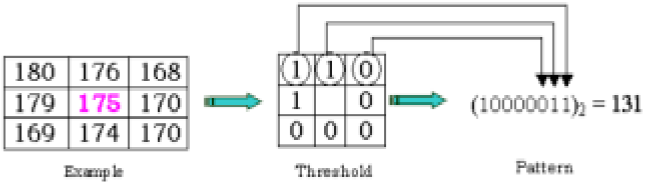


Figure 1.4: Example of LBP calculation.

Probably the most prominent and most frequently addressed application of Image Features is detecting and recognizing objects in images. Recently, several systems have been presented, which are able to detect and recognize objects from several viewing directions in cluttered scenes with high accuracy.

1.4 Object Recognition

The term object can be used in a very broad manner, so it’s difficult to give a formal definition. An object can be considered as a part of, or token in, a sensory signal. The precise representation of the object within the signal can undergo changes such as scaling, translation, or other deformations, or it can be contaminated by noise or be partially occluded. These changes give rise to an entire collection or class of signals which can all still be associated with the original object. Objects in signals often correspond to physical objects in the real world environment from which the signals have been recorded. This is the case for objects in images of natural scenes. In physics, a physical body or physical object (sometimes simply called a body or object) is a collection of masses, taken to be one. For example, a football can be considered an object but the ball also consists of many particles. Classes of objects are collections of objects that are similar.

Object detection and recognition has attracted significant attention over the past few years in the field of computer vision, pattern recognition and image processing [1-3]. ***Object detection*** approach first came into existence in 1974 by Yoram Yakimovsky, who provided automatic location of objects in digital images [2]. It is a process of detection and recognition of certain classes like chairs, guitars, buildings etc in image or video sequence. Humans can recognize objects in images or videos with little effort, despite the fact that the image can contain objects of many different sizes/scales or they can be translated or rotated and can vary in different viewpoints [5-7]. Various researches have been done till now for detection of objects from real-world images [2,18,22], objects from noisy images [3,24], objects from videos [20,23], objects moving in space [1] and also tracking multiple objects [19]. Various approaches like feature classification[18], combination of appearance, structural and shape features [19], viewpoint independent object detection algorithm [23], wavelet transform methods[24] were adopted for efficient object detection. ***Object recognition*** can be described as artificial intelligence and subfield of computer vision research. Application of artificial intelligence is Pattern recognition. Object recognition system might have to embody or develop some understanding of the concepts in order to identify objects of the class. The problem in object recognition is to determine which, if any, of a given set of objects appear in a given image or image sequence. Thus object recognition is a problem of matching models from a database with representations of those models extracted from the image luminance data. Objects are recognized from many different vantage points (from the front, side, or back), in many different places, and in different sizes. Objects can even be recognized when they are partially obstructed from view.

The problems of object detection and object recognition are closely related. An object recognition system can be built out of a set of object detectors, each of which detects one object of interest. Similarly, an object detector can be built out of an object recognition system; this object recognizer would either need to be able to distinguish the desired object from all other objects that might appear in its context, or have an unknown object class. Thus the two problems are in a sense identical, although in practice object recognition systems are rarely tuned to deal with arbitrary backgrounds, and object detection systems are rarely trained on a sufficient variety of objects to build an interesting recognition system. The different focuses of these problems lead to different representations and algorithms. In this work, we will concentrate on object recognition.

Object recognition in the field of computer vision focuses on the task of identifying or locating a set of objects in an image or video sequence. Object Recognition is one of the most important, yet least understood aspects of visual perception. In contrast to methods of specific object (say, face, fingerprint, fish, aircraft etc.) recognition, which endeavour to distinguish between multiple objects within a class; Generic Object Recognition (GOR) spans pose-invariant recognition of multiple classes of objects having a wide variety of distinguishing features. Thus, the problem is not restricted to a single class of objects, say only face recognition or vehicle recognition. Rather, it involves object recognition from novel views of multiple categories of objects for which some training samples are available. Recognition domains where the exact features distinguishing one class of objects from others are unknown have revived our interest in generic object recognition. There are two main choices for the object recognition strategy: the feature-based strategy, which is based on shape information and the image-based strategy, which is based on direct representation of image intensity or on a filtered version of the image [91].

- **Feature-based approach:** This computational strategy for object recognition is based on the idea that much of the information about an object is encapsulated by its geometrical properties. It usually relies on a geometrical model of an object’s shape characteristics which is often applied to simple data, and is used to explore the correspondences between the model’s features and the detected features in the scene during recognition. Given an unknown scene and an object model, both represented in terms of their features, in this approach the objective is to find a partial match between the two and estimate the object’s location and pose in the image. The limitation here is that object's features in the image must be consistent with some pose of the object, so transformation space is considered to overcome this limitation. This strategy is quite inefficient since a large number of views must be stored for each model, unless we utilize some of the techniques. In this approach first an object is detected from an image then its feature matrix is calculated. Effectiveness of features and efficiency of a matching technique must be considered when choosing an object recognition strategy.

- **Image-based approach:** A desirable characteristic of image-based recognition is that object models can be compared directly or fairly directly with input data, as both are of the same type (e.g. intensity images). Feature-based methods instead require that features be detected and described before data and model can be compared. Instead an image-based approach does not need to recover the geometry of the objects but can learn their appearance characteristics from training imagery. A model of the object is built off-line from a collection of different images depicting a variety of object appearances taken under changing viewpoints and lighting conditions. In this way, each model view is stored as a vector of image intensities in some low-dimensional space that captures the significant characteristics of the object, such as the eigenspace. Recognition is carried out by projecting the image of an object to a point in the low-dimensional space. The object is recognized by calculating the shortest distance from a given models. Other image intensity methods include use of color histograms and photometric invariants and eigenspace methods and many more. Image based methods can thus be successful in handling the combined effects of shape, pose, reflection and illumination, but have serious difficulties in segmenting the object(s) from the scene and dealing with occlusions. Since matching is performed directly in the image domain, rather than in the geometric feature domain, performance is not affected by increasing geometric complexity. A great advantage of image-based methods is that any shape can be represented no matter how complex as long as we can take images of it. Image-based was considered superior in object recognition performance and simpler in use. The feature-based strategy, however, may allow a higher recognition speed and smaller memory requirements.

Some of the approaches of object recognition are explained in detail in next section. After that some methods are compared. Then some challenges faced by object recognition approaches are explained and finally some applications of object recognition are mentioned.

1.4.1 Approaches of feature extraction for object recognition

The computer vision literature is rich with approaches in solving the object recognition problem and based on the applied features these methods can be subdivided along many different lines on the basis of architecture point of view. So the various approaches of feature extraction for object recognition are:

1.4.1.1 Appearance-based approaches

Appearance–based approach has been directly based on images and extracts features corresponding to a particular appearance of the object, which is usually captured by different two-dimensional views of the object-of-interest. So representations of objects, which use only information of images, are called appearance based models. This approach is encouraged due to its robustness, speed and success in recognizing objects. Appearance-based object recognition methods have recently shown good performance on a variety of problems. Objects look different under varying conditions such as changes in lighting or color, changes in viewing direction, changes in size / shape. However, many of the methods either require good whole-object segmentation, which severely limits their performance in the presence of clutter, occlusion, lighting, changes in orientation or background changes; or utilize simple conjunctions of low-level features, which cause crosstalk problems as the number of objects is increased. This approach extracts features such as complex, curved shapes. However, it is impossible to represent all appearances of an object.

In the classical approach to image analysis approach image features such as edges or image regions such as texture regions are extracted from the image. High level feature groups can be obtained by grouping these basic image features. The principal difficulties with this classical approach are that the process of determining feature is complex; image features extracted are unstable, broken and spurious. In order to make the problem tractable, the number of extracted features must be reduced. This implies the use of salient -- meaning discriminant -- features. Because of the exponential complexity, a relatively small number of image features can be used so that each image feature must be highly discriminant. Due to the trade-off between robustness of the feature extraction and the discriminant power of features, the process of feature extraction becomes unstable and these techniques are suitable only for particular object classes such as geometric objects [92]. The above limitations of the classical approach to image analysis can be removed by the model of an object whose representation is given by image measurements which can be learned automatically from sample images. These techniques are called appearance based methods since each of the represented images corresponds to a particular appearance of the object. The advantage of appearance based methods is that they can use robust image measurements and that they can avoid feature correspondence. The speed and the robustness of appearance based object recognition approaches comes with a price as appearance based approaches use directly image measurements for recognition. During the last few years, there has been a growing interest in object recognition directly based on images, each corresponding to a particular appearance of the object. Representations of objects, which only use 2D--information of images are called appearance based models. The benefit of such representation schemes is most obvious in areas like face recognition, human--computer interfaces and content based image retrieval. The main challenges are the recognition of objects in the presence of partial occlusion, the recognition of 3D objects and the classification of objects [92]. These challenges have been handled up to some extent like some systems demonstrates extraordinarily good recognition for a variety of 3-D shapes, ranging from sports cars and fighter planes to snakes and lizards with full orthographic invariance by utilizing distinctive intermediate-level features i.e. by automatically extracting 2-D boundary fragments, as keys, which are then verified within a local context, and assembled within a loose global context to evoke an overall percept. The method is robust to occlusion and background clutter, and does not require prior segmentation. In the earlier system, local features based on automatically extracted boundary fragments are used to represent multiple 2-D views (aspects) of rigid 3-D objects, but the basic idea could be applied to other features and other representations [93].

Appearance-based methods span a wide spectrum of algorithms, which can be roughly classified into global and local approaches. *Global approaches* model the information of a whole image which is done by subspace methods. In contrast to that, local approaches search for salient regions characterized by e.g. corners, edges or entropy. *Local approaches* recognize and localize objects on the base of local features. In a later stage, these regions are characterized by a proper descriptor. For object recognition purposes the thus obtained local representations of test images are compared to the representations of previously learned training images. Several methods have been proposed for feature detection, among which the most popular is the Harris corner detector [2], Shi-Tomasi features, SIFT features, and Maximally Stable Extremal Regions [85].

1.4.1.1.1 Edge matching: An edge may be regarded as a boundary between two dissimilar regions in an image, which may be different surfaces of the object, or perhaps a boundary between light and shadow falling on a single surface. Most edge detection methods work on the assumption that an edge occurs where there is a discontinuity in the intensity (or depth) function or a very steep intensity (or depth) gradient in the image. Changes in lighting and color usually don't have much effect on image edges. All edge detection algorithms try to locate the edges in an image presented by true (real) scene elements but not noise. They can be classified into six categories [90]:

1. ***First Order Derivative:*** The basic idea is to locate the position where the first derivatives are local maxima as edge/s. Various operators used are Simple operator, Robert's cross operator (use the diagonal directions to calculate the gradient vector), Prewitt operator (detect vertical and horizontal edges), Sobel operator (this is relatively easy to implement in hardware form, most obviously by a pipeline approach), Kirsch Operator (considering edge direction more accurately, this operator uses 8 convolution masks).
2. ***Second Order Derivative:*** The basic idea is to find the pixels in an image where zero-crossings are treated as edges. This is based on the observation that a maximum of the first derivative occurs at a zero crossing of the second derivative. Various operators used are Laplacian operator (linear and rotationally symmetric), LoG (Laplacian of the Gaussian) (Marr-Hildreth Operator).
3. ***Canny Edge Detector*** treats edge detection as a signal processing problem and tries to design a "optimal" edge detector.
4. ***Model Fitting:*** The basic idea is to develop an edge model and then to compute its degree of match to the image. An edge is modeled by specifying its four degrees of freedom: its position, its orientation, and the constant intensities on either side of the step. The model is located to match the data by simply seeking the least squares error fit of the parametric model to the image window, an edge is declared present when the weighted least squares error is smaller than some pre-set threshold.
5. ***Phase Congruency:*** Many kinds of features can be detected using phase congruency, a frequency-based approach. Edges are detected at points where the Fourier components of a waveform come into phase. At these points the local energy is maxima. The edges will be found searching specific values of phase congruency between the points of local maxima energy.
6. ***Regularization-based:*** This approach based on MRF (Markov Random Field) theory to set up a cost function for edge existence, then MAP (Maximum a priori) estimates the occurrence of edges.

1.4.1.1.2 Gradient matching: Another way to be robust to illumination changes without throwing away as much information is to compare image gradients. Matching is performed like matching grayscale images. Simple alternative is using (normalized) correlation.

1.4.1.1.3 Corners: Image corner detection is an important task in various computer vision and image understanding systems, because corners are proven to be stable across sequences of images. Applications include motion tracking, object recognition, and stereo matching. The feature formed at boundaries between only two image brightness regions, where the boundary curvature is "extremely" high. The corner finder is based on the fact that each point within an image has associated with it a local area of comparable brightness. Broadly speaking, there are three approaches for corner detection [90].

1. ***Edge-based:*** Based on pre-segmented contours. A few methods using binary edge maps to find corners where first edges are detected and then edge curvature is calculated in order to find corner locations.
2. ***Raw-based:*** Directly based on the differential analysis of raw gray-scale images. Template-based detects the similarity between a given templates of a specific angle for each image sub-window. Gradient-based uses the gradient differential of image directly.
3. ***Combine edge and curvature:*** Rely on measuring the curvature of an edge that passes through a neighborhood. Methods using the product of the gradient magnitude and the edge contour curvature (curvature is a local measure of how fast a planar contour is turning). Many other methods were used to achieve the same.

Local appearance based object recognition systems work on distinguished regions in the image, it is of great importance to find such regions in a highly repetitive manner. If a region detector returns only an exact position within the image we also refer to it as interest point detector (we can treat a point as a special case of a region). Ideal region detectors deliver additionally shape (scale) and orientation of a region of interest. The currently most popular distinguished region detectors can be roughly divided into three broad categories: corner based detectors, region based detectors, and other approaches. ***Corner based detectors*** locate points of interest and regions which contain a lot of image structure (e.g., edges), but they are not suited for uniform regions and regions with smooth transitions. ***Region based detectors*** regard local blobs of uniform brightness as the most salient aspects of an image and is therefore more suited for the latter. ***Other approaches*** for example take into account the entropy of a region (Entropy Based Salient Regions) or try to imitate the human's way of visual attention. Feature descriptors describe the region or its local neighbourhood already identified by the detectors by certain invariance properties. Invariance means, that the descriptors should be robust against various image variations such as affine distortions, scale changes, illumination changes or compression artifacts (e.g., JPEG). It is obvious, that the descriptors performance strongly depends on the power of the region detectors. Wrong detections of the region's location or shape will dramatically change the appearance of the descriptor. Nevertheless, robustness against such (rather small) location or shape detection errors is also an important property of efficient region descriptors. One of the simplest descriptors is a vector of pixel intensities in the region of interest. Roughly speaking descriptors can be divided into three main categories i.e. distribution based descriptors, filter based descriptors and other methods. *Distribution-based methods* represent certain region properties by (sometimes multi-dimensional) histograms. Very often geometric properties (e.g., location, distance) of interest points in the region (corners, edges) and local orientation information (gradients) are used. The descriptors in this category are SIFT [87]; PCA-SIFT (gradient PCA); gradient location-orientation histograms (GLOH), sometimes also called extended SIFT [?90]; Spin Images; shape context [19] [90] and Locally Binary Patterns [94]. *Filter based methods* are differential-invariants and complex and steerable filters. *Other methods* include cross-correlation and moment-invariants [85]. Some of the approaches were explained earlier.

1.4.1.1.4 Moment Invariants: Generalized intensity and color moments have been introduced by Van Gool in 1996 to use the intensity or multi-spectral nature of image data for image patch description. The moments implicitly characterize the intensity (I), shape or color distribution (R, G, B are the intensities of individual color components) for a region Ω and can be efficiently computed up to a certain order and degree. Combinations of such generalized moments are invariant to geometric and photometric changes. Combined with powerful, affine invariant regions based on corners and edges they form a very powerful detector-descriptor combination [85].

1.4.1.1.5 Shape Context: Shape context descriptors have been introduced by Belongie et al. in 2002. The goal is to find a shape which corresponds to the shape of the reference object. Due to changes in viewpoint the shape has to be described in a scale and rotation invariant form. In their original form, shape context features characterize shape in 2D images as histograms of edge pixels. They use the distribution of relative point positions and corresponding orientations collected in a histogram as descriptor. The primary points are internal or external contour points (edge points) of the investigated object or region. The contour points can be detected by any edge detector, e.g., Canny- edge detector, and are regularly sampled over the whole shape curve. A full shape representation can be obtained by taking into account all relative positions between two primary points and their pair-wise joint orientations. It is obvious that the dimensionality of such a descriptor heavily increases with the size of the region. To reduce the dimensionality a coarse histogram of the relative shape sample points coordinates is computed i.e. the shape context. The bins of the histogram are uniform in a log polar2 space (see Figure 1.5) which makes the descriptor more sensitive to the positions nearby the sample points [85].

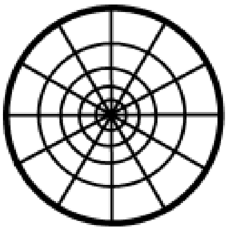


Figure 1.5: Histogram bins used for shape context.

It has been shown that powerful object recognition systems can be built on the base of local features. However, there are many cases where objects do not contain enough features such as objects in our kitchen environment. For example, some dishes contain real texture features, since they are often solid colored with only very few textural information. For such objects, it is more sensible to assume the objects can be segmented, e.g. by color, and solve the problem of recognition and localization with a global appearance-based approach. In global appearance-based methods for object recognition, the discussion is reduced to subspace methods whose main idea is to project the original input images onto a suitable lower dimensional subspace that represents the data best for a specific task. Global appearance-based can be done by linear and non linear analysis [89].

Linear Analysis: Three classical linear approaches are introduced here, which are Principal Component Analysis (PCA) [96], Independent Component Analysis (ICA) [97] and Linear Discriminant Analysis (LDA) [82]. By projecting a face image to a subspace, the projection coefficients are used as the feature representation of each face image. The classification is performed between the test face image and the training prototype. All representations from these three linear approaches are considered as a linear transformation from the original image vector x to a projection feature vector y, i.e. $y = W^T x$, where W indicates the transformation [15].

1.4.1.1.6 Principal Component Analysis: Principal Component Analysis (PCA) [96] is a powerful appearance based feature extraction technique and widely used technique in statistics, which applies the Karhunen-Loeve transform to a set of training images and derives a number of projection axes that act as the basis vectors for the PCA subspace. PCA is a linear transformation aimed at minimizing the representation error, widely used for face representation and face recognition. All face images in the training set are collected and composed into a covariance matrix. The eigenvectors and corresponding eigenvalues are generated from the covariance matrix. The eigenvectors are visualised into a two-dimensional array and display ghost face like appearance. Hence, the eigenvectors are also named "eigenfaces". These eigenfaces span a small subspace in the image space. The subspace is called "face" subspace. Given the eigenfaces, every face in the training set can be represented as a vector with a sequence of weights. The weights are obtained by projecting the images into the face subspace by an inner product operation. The testing image is also represented by its vector. Each face image is represented as a vector of projection coefficients in this subspace, in which information compression, dimensionality reduction and de-correlation of the pixel values of the face images is achieved. The mapping is defined by a set of basis vectors which correlate to the maximum variance directions present in the training data. Hence, each training image can be projected into this subspace and again reconstructed with minimum error [15]. Global PCA (principal component analysis) based methods are sensitive to variations in the background behind objects of interest, changes in the orientation of the objects, and to occlusion [89].

1.4.1.1.7 Linear Discriminant Analysis: Linear Discriminant Analysis (LDA) is a statistical method to classify examples into different classes based on a set of measurements of examples. The LDA applied in face recognition is very successful, because LDA is originally for classification, i.e., the LDA is a supervised learning approach. In LDA, the purpose is to maximise the discrimination between different classes, and recognition can be apparently done based on this. The LDA also constructs a subspace that is constructed by the selected components. The LDA training is carried out by using scatter matrices. The method selects a set of features in such a way that the ratio of the between-class scatter and the within-class scatter is maximised. The linear discriminant analysis (LDA), also called Fisher's LDA takes into account the class information in feature reduction. It tries to simultaneously maximize the distances between the class centers and to keep the distances within one class constant. This can be achieved using within-class and between-class scatter matrices, leading to a generalized eigenvalue problem [15]. Unlike PCA, which considers only the variance of the training images to construct a subspace, linear discriminant analysis (LDA) aims at improving upon PCA by also taking the class-membership information of the training images into account when seeking for a subspace and aims at maximizing the separability of the classes, which is usually wanted in pattern recognition. From this point of view PCA is usually considered as being more appropriate for the task of data compression, while LDA is tailored more towards the classification task [89]. Moreover, there are several extensions of the standard approach such as robust classification, incremental LDA, or 2D representations.

1.4.1.1.8 Independent Component Analysis: Independent Component Analysis (ICA) was originally introduced by Herault, Jutten, and Ans in the field of neurophysiology [97]. But it became widely known and popular method not until it was introduced in signal processing for blind source separation, i.e., separation of mixed audio signals. This problem is often described by the task of identifying a single speaker in a group of speakers ("cocktail-party problem"). Independent component analysis (ICA) [97] is an extension of PCA, which in addition to second-order statistics present in the training images tries to minimize higher-order dependencies in the training images as well. It does so by seeking basis vectors along which the projected images are statistically independent. Unlike PCA and LDA, which represent orthogonal transformations (i.e., the basis vectors comprising the transformation matrices are orthogonal one to another), ICA represents a non-orthogonal transformation. Like the PCA, Independent Component Analysis (ICA) is also a linear transformation. The PCA is to find the ranked principle components which describe the variation, however the ICA is to find the independent components by maximising the statistical independence between these estimated components. The independence of the components is measured by maximising Non-Gaussian distribution. The popular algorithms to perform ICA include Infomax, FastICA, and JADE. Two different architectures of ICA have been proposed for the purpose of face recognition. The first, commonly referred to as ICA Architecture I (ICA1) seeks statistically independent basis vectors, while the second, commonly referred to as ICA Architecture II (ICA2) seeks statistically independent features (or projection coefficients). Both ICA architectures can be implemented by subjecting either the transformation matrix of the PCA technique or the PCA feature vectors corresponding to the training images to the FastICA algorithm.

Non-Linear Analysis: In non-linear analysis, it is assumed that objects reside in non-linear structures. The non-linear structures are normally described as manifold [15]. A manifold is an abstract mathematical space in which every point has a neighbourhood which resembles a Euclidean space, but in which the global structure may be more complicated. In a one-dimensional manifold (or one-manifold), every point has a neighbourhood that looks like a segment of a line. Examples of one-manifolds include a line, a circle, and two separate circles. In a two-dimensional manifold, every point has a neighbourhood that looks like a disk. Examples include a plane, the surface of a sphere, and the surface of a torus. Manifolds are important objects in mathematics and physics because they allow more complicated structures to be expressed and understood in terms of the relatively well-understood properties of simpler spaces. Kernel methods have been successfully applied to solve pattern recognition problems because of their capacity in handling nonlinear data. By mapping sample data to a higher dimensional feature space, effectively a nonlinear problem defined in the original image space is turned into a linear problem in the feature space. PCA or LDA can subsequently be performed in the feature space and thus Kernel Principal Component Analysis (KPCA) and Generalized Discriminant Analysis (GDA) [89].

1.4.1.1.9 Kernel PCA, Kernel LDA, Kernel ICA & ISOMAP Embedding

- **Kernel PCA:** As PCA has become one of the most successful approaches in feature extraction. However, PCA only uses the second order statistical information in data. As a result, it fails to perform well in nonlinear cases. Kernel PCA (KPCA) is able to capture the nonlinear correlations among data points, and in some cases has been more successful than conventional PCA. Yanmei Wang and Tang, propose a method of feature extraction based on KPCA. Kernel principal component analysis is a method of non-linear feature extraction. Here nonlinearly separable patterns in an input space will become linearly separable with high probability if the input space is transformed nonlinearly into a high-dimensional feature space. Thus, therefore, map an input variable into a high-dimensional feature space, and then perform PCA. Because computing a covariance matrix is based on dot product, performing PCA in the high-dimensional feature space can obtain high-order statistics of the input variables, that is, also the initial motivation of the KPCA. However, it is difficult to directly compute both the covariance matrix and its corresponding eigenvectors and eigenvalues in the high-dimensional feature space. It is computationally intensive to compute the dot products of vectors with a high-dimension. Fortunately, kernel tricks can be employed to avoid this difficulty, which computes the dot products in the original low-dimensional input space by means of a kernel function.

- **Kernel LDA:** In statistics, Kernel LDA also known as kernel Fisher discriminant analysis (KFD), generalized discriminant analysis (GDA) and kernel discriminant analysis, is a kernelized version of linear discriminant analysis (KLDA). It is named after Ronald Fisher. Using the kernel trick, LDA is implicitly performed in a new feature space, which allows non-linear mappings to be learned. The Kernel Fisher Linear Discriminant (KFLD) is similar to KPCA. The projected examples are centred in the feature space. The within-class and between-class scatter matrices are calculated then applying LDA in kernel space. Similar to LDA, the purpose of KLDA is to maximize the quotient between the inter-class inertia and the intra-classes inertia. Thus, rather than explicitly mapping the data to new dimension, the data can be implicitly embedded by rewriting the algorithm in terms of dot products and using the kernel trick in which the dot product in the new feature space is replaced by a kernel function as was in KPCA.
- **Kernel ICA:** The kernel based ICA is proposed by Bach and Jordan, who use contrast functions based on canonical correlations in a reproducing kernel Hilbert space to develop a new class of ICA algorithm. The Kernel ICA algorithm is based on the minimization of a contrast function based on kernel ideas. A contrast function measures the statistical dependence between components, thus when applied to estimated components and minimized over possible de-mixing matrices, components that are as independent as possible are found.
- **ISOMAP Embedding:** Isometric Feature Mapping (ISOMAP) is a global optimal and asymptotic method in which convergence guarantees the flexibility to learn a broad class of nonlinear manifolds. The approach seeks to preserve the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The core is to estimate the geodesic distance between far away points, given only input-space distances. For neighbouring points, input space distance provides a good approximation to geodesic distance. For faraway points, geodesic distance can be approximated by adding up between neighbouring points. These approximations are computed efficiently by finding shortest path in a graph with edges connecting neighbouring data points. Figure 1.6 describes how ISOMAP exploits geodesic paths for nonlinear dimensionality reduction on the "Swiss roll" data set. In Figure 1.6(A), there are two arbitrary points (circled) on a nonlinear manifold. The Euclidean distance between them in the high dimensional input space may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve).

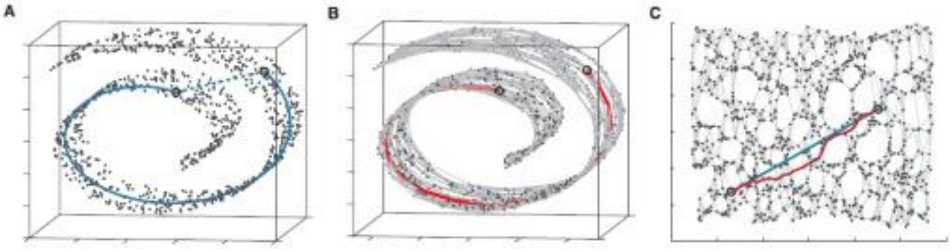


Figure 1.6: ISOMAP exploits geodesic paths on the “Swiss roll” dataset.

In Figure 1.6(B), the neighbourhood graph G constructed in ISOMAP with k-near neighbour with k = 7 on 1000 data points. It allows an approximation (red segments) to the true geodesic path to be computed. Figure 1.6(C) shows the two-dimensional embedding recovered by ISOMAP, which best preserves the shortest path distances in the neighbourhood graph. Straight lines in the embedding represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph path (red lines).

1.4.1.2 Model-based approach

In model- (or shape-, or geometry-) based methods, the information about the objects is represented explicitly. The recognition can then be interpreted as deciding whether (a part of) a given image can be a projection of the known (usually 3D) model of an object. Generally, two representations are needed: one to represent object model, and another to represent the image content. To facilitate finding a match between model and image, the two representations should be closely related. However, the model and image representations often have distinctly different “meanings”. The model may describe the 3D shape of an object while the image edges correspond only to visible manifestations of that shape mixed together with ”false” edges and illumination effects (shadows) [89]. Model-Based object recognition in real-world outdoor situations is difficult because a robust algorithm has to consider multiple factors such as: i) object contrast, signature, scale, and aspect variations; ii) noise and spurious low resolution sensor data; and iii) high clutter, partial object occlusion, and articulation. Current approaches use shape primitives, contours, colors, and invariant object features for matching. The performance of these methods is acceptable when objects are well defined, have high contrast, and are at close ranges. To improve the recognition performance under multi-scenarios and varying environmental conditions, model of sensors, atmosphere, and background clutter are helpful in addition to the geometric model of an object. Using only a minimum set of object models and sensor model, multi-scale Gabor wavelet representation of objects and a flexible matching mechanism described in this paper can potentially help to improve the recognition performance under real-world situations.

1.4.2 Comparison of object recognition approaches

Object recognition is another research area in computer vision and image processing. Generally speaking, it is a method to match the features of a given object against those of some predefined object samples. Object recognition was done by many methods like Pattern matching, principal components analysis method (PCA), graph matching, General Hough transform (GHT) [28], Wavelet packet and Gabor wavelet. These methods can be enhanced to a three-dimensional representation as shown in [7].

Pattern matching approaches [28] are widely used due to their simplicity. It uses squared differences between a camera image and a template and is summed pixel-wise. This approach is not suitable for occlusions of objects, the recognition rates are changed in an unacceptable way due to light reflected by metallic surfaces. Additionally, slight rotations can hardly be handled. Object recognition by PCA [28] is a correlation based technique. Every object is segmented from the background, which is scaled and then normalized. PCA is done with eigenvectors. The main drawback of PCA is the sensitivity. As PCA is a correlation based technique, there are problems with object occlusions. When the image size, position, orientation or illumination changes even slightly, the PCA system fails. If parts of the object are segmented or the object area is actually bigger than the object, the PCA tends to suffer. The graph matching method [28] deal with the idea to walk along orthogonal lines that pass certain sampling points on a contour model. A great disadvantage of this approach is that variety of parameters that have to be tuned. Small objects are difficult to find, also hard in finding the exact rotational angle of symmetric objects. The GHT [28] is an extension of the original Hough Transform. The contour model is approximated by a set of sample points. Every point can be described with respect to some reference point inside the contour through the vector. The general disadvantage of GHT is it is not suitable for small objects that can hardly be distinguished, object detection becomes difficult with cluttered background and high memory consumption.

Gabor wavelet had been used in the past for object detection in Infrared images [38], 3D object recognition in [7,40], and object tracking in [39]. The main aim to use Gabor wavelets is due to their multi-resolution, multi-orientation properties. The use of Gabor wavelet approach has several advantages such as robustness against facial expression, illumination, facial hair, glasses, image noise and invariance to some degree with respect to small changes in head pose, selectivity in scale, as well as selectivity in orientation. Further advantages include: More efficient and accurate compare to traditional approaches, saves neighbourhood relationships between pixels, easy to update, fast recognition and low computational cost [29]. Gabor wavelets are used for extracting local features for various applications such as object detection, recognition and tracking [5,6], face tracking [31], optical character recognition, iris recognition, fingerprint recognition, and texture analysis as well in neurophysical studies [30].

1.4.3 Challenges in Object Recognition

Object recognition has proven to be a significantly difficult challenge especially with the complexity of real world data, in which there is great variation in both the appearance of objects within a single object class (e.g., mugs may come in many shapes and colors), and in the appearance of the same object under various circumstances (e.g., the same object can appear different with changes in pose, size and lighting). The varying appearances and circumstances can extremely change the pixel values in an image for the same object. So variations in geometry, photometry and viewing angle, noise, occlusions and incomplete data are some of the problems with which object recognition systems faced. Static-image object recognition, as well as some video-based ones, has dealt with this complexity by focusing on learning robust visual features. Successful approaches to object recognition addresses a variety of problems, some explained here are [73]:

- **Changes of aspect:** Different views of an object can look very different (see Figure 1.7).

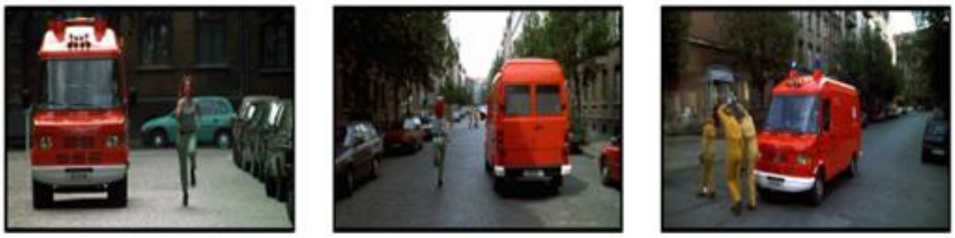


Figure 1.7: Variation due to changes in aspect.

- **Changes of viewpoint:** Objects can also be subject to in-plane transformations (translation, rotation, scaling, and skews) and out-of-plane transformations (foreshortenings) that change their appearance. However, some viewpoints may be more likely than others (i.e. motorbikes are rarely vertically orientated).
- **Illumination differences:** A change in the lighting of the object will change the pixel values in the image. The change could be a shift or scaling of the pixel values or, if the light source changes position, a non-linear transformation, complicated by shadows falling on the object. The images in Figure 1.8 illustrate examples of drastic changes in lighting.

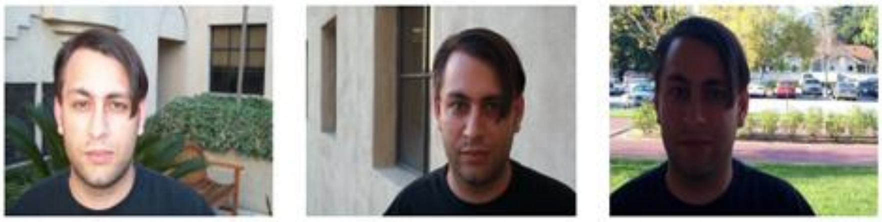


Figure 1.8: Variation in appearance due to a change in illumination.

- **Background clutter:** In the majority of images it is rare for the object to be cleanly segmented from the background. More typically the background of the image contains many other objects (other than the one of interest), which distract from the object itself. The images in figures 1.7 and 1.8 have cluttered backgrounds.

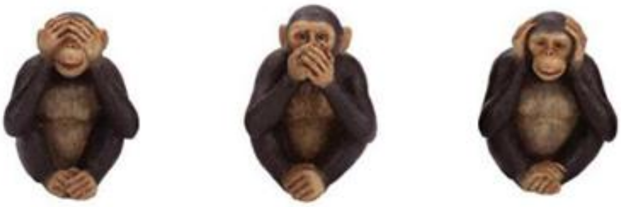


Figure 1.9: Some examples of occlusion.

- **Occlusion:** Some parts of the object may be obscured by another object, as illustrated by the monkeys in figure 1.9. Additionally, as the aspect changes, one part of the object may hide another. This is known as self-occlusion.

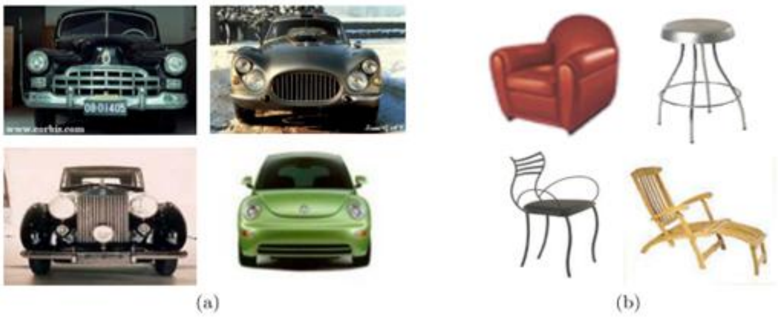


Figure 1.10 (a): Some examples from a visual category. **(b):** Some examples from a functional category.

- **Intra-class variation:** As in the car example of figure 1.10(a), the category itself can have a large degree of visual variability. The variability can take various forms: in the geometry, appearance, texture and so on. Also, one instance of an object may have features that are missing on another (e.g. the radiator grille on the cars of figure 1.10(a)).

Object category is of two types: visual and functional. In the former, the objects are related by some kind of visual consistency, be it in the outline or the appearance of the object. In contrast, a functional category is one that is related by its purpose, for chairs, as shown in

Figure 1.10(b). Functional categories cannot be modelled using visual information alone and since the methods in this thesis use only visual cues, we limit ourselves to visual categories.

1.4.4 Applications of Object Recognition

Object recognition has been used in several application fields, in high-definition video [26], for high-resolution satellite images [25], in driver assistance systems [27], for programming by demonstration applications [28]. Object recognition methods have numerous applications, from which some of the following are [43]:

- **Automatic Target Detection and Recognition**: Automatic Target Recognition (ATR) generally refers to the autonomous or aided target detection and recognition by computer processing of data from a variety of sensors. It is an extremely important capability for targeting and surveillance missions of defence weapon systems operating from a variety of platforms. The major technical challenge for ATR is contending with the combinatorial explosion of target signature variations due to target configuration and articulation, target/sensor acquisition geometry, target phenomenology, and target/environment interactions. ATR systems must maintain low false alarm rates in the face of varying and complex backgrounds, and must operate in real time. The main objective is to locate and identify time-critical targets and vehicles of military interest to aid in surveillance operations, battlefield reconnaissance, intelligence gathering, remote sensing, weapons guidance, and exploitation of imagery from unmanned aerial vehicles and other reconnaissance platform. A second application is to look for militarily significant change detection, site monitoring, battle damage assessment and activity tracking. An operational goal is to significantly reduce the volume of imagery presented to a human image analyst. The ATR field evolves from using statistical pattern recognition approaches to model-based vision, recognition theory, and knowledge-based information exploitation systems.

- **Autonomous Robots:** Any mobile robot needs to sense its environment to maintain a dynamic model of the external world and develop an intelligent computational capability for visual processing. The visual analysis of shape and spatial relations is used in many other tasks, such as object manipulation, planning, grasping, guiding and executing movements in the environment, selecting and following a path, or interpreting and understanding world properties. Modern home and service robots work in complex environments with complex objects and are able to perform a variety of tasks both indoors and outdoors.

- **Vehicle navigation and obstacle avoidance:** It includes mobile robots as well as autonomous vehicles, smart weapons, and unmanned platforms that navigate through an unknown or partially-known environment. Research in this field has received considerable attention in past two decades due to the wide range of potential applications, from surveillance to planetary exploration. Autonomous vehicle control, symbolic planning and environment exploration involve the actions of moving around, tracking objects of interest, planning a safe route to avoid collision, serving to guide motion with respect to road constraints, and integrating sensor information for efficient navigation.
- **Industrial Visual Inspection:** The traditional examples of object recognition come from the domain of visual inspection of industrial parts. Among the numerous applications we can mention the following: assembly control and verification, metrology, precision measurements of machine parts and electronic patterns, unconstrained material handling, geometric flaw inspection, surface scan and assembly, food processing, quality control, manufacturing, modelling and simulation.
- **Face Recognition:** People in computer vision and pattern recognition have been working in automatic recognition of human faces for more than 25 years. Recently there has been renewed interest in the problem due in part to numerous security applications ranging from identification of people in police databases to video-based biometric person authentication, and identity verification at automatic teller machines. Numerous commercial systems are currently available. The potential applications include, but are not limited to: video surveillance and monitoring, building security, site monitoring, videoconferencing, law enforcement operations, photo interpretation, medical, commercial and industrial vision.
- **Medical Image Analysis:** Medical image analysis has developed into an independent flourishing branch of computer vision and image processing as is evidenced by the tremendous interest and growth in the field. Medical imaging concerns both the analysis and interpretation of biomedical images through quantitative measurements and manipulation of objects, and the visualization of qualitative pictorial information structures. The main purpose of current research in medical imaging is to improve diagnosis, evaluation, detection, treatment and understanding of abnormalities in internal physiological structures and in their function. The last decades have witnessed a revolutionary development in the use of computers and image processing algorithms in

the practice of diagnostic medicine. Images of both anatomical structure and physiological functioning are now produced by a host of imaging modalities: computerized tomography, magnetic resonance imaging, optical sectioning, positron-emission tomography, cryosectioning, ultrasound, thermography and others. This has enabled the acquisition of detailed images carrying vast amounts of multidimensional information. Furthermore, we have seen the appearance and dissemination of online digital libraries of volumetric image data, such as the "Visible Human" project, undertaken by the U.S. National Library of Medicine, which comprises the construction of highly detailed templates for human anatomies from various digital sources. Medical Imaging is one of the most dynamic research fields in action today.

- **Optical character recognition:** The importance of document image analysis and optical character recognition has increased markedly in recent years, since paper documents are still the most dominant medium for information exchange, while the computer is the most appropriate device for processing this information. Document image understanding and retrieval research seeks to discover efficient methods for automatically extracting and organizing information from handwritten and machine-printed paper documents containing text, line drawings, graphics, maps, music scores, etc. Its characteristic problems include some of the earliest attempted by computer vision researchers. Document analysis research supports a viable industry stimulated by the growing demand for digital archives, document image databases and paperless sources, the proliferation of inexpensive personal document scanners, and the ubiquity of fax machines. Related areas of research include document image databases, information filtering, text categorization, hand-written document interpretation, document image understanding and retrieval, etc.
- **Web search:** Internet image search engines are currently the only way to find images on the Internet. Their poor performance is due to their use of the image filename or surrounding HTML rather than the actual image content. The natural way to find images is to search visually. As it is difficult to specify a visual query directly, the user might select a few examples, similar in nature to the desired image. From these a visual model would be trained that could then be used to perform the search. Such a scheme would require the ability to learn a model quickly from a few examples [73].
- **Searching image databases:** Many people have thousands of home photos on their computers which are only loosely organized. Without manual annotation, the images

cannot be searched to find all instances of the family dog, for example. Many companies also have large archives of images which they wish to search.

- **Online dating:** Online dating websites only have the facility to search for people based on information manually entered by people (e.g. hair color, eye color). A more useful form of search would utilize the photo placed online by each person subscribing to the website. For example, on finding an attractive person, you could search for people that looked like him or her. This requires building models of people that you (or people with similar tastes to you) find attractive.
- **Security:** Looking for people or vehicles in video streams from security cameras. Many major cities have thousands of close-circuit TV cameras whose footage must be manually scrutinised. Automated surveillance systems do not get tired or distracted unlike humans who currently monitor such footage.
- **Airport baggage screening:** People currently examine airplane passenger luggage using x-ray machines. The large number of bags examined per hour and the difficulty in interpreting the x-ray images mean the false negative rate is high. Recognition systems designed to find knives and guns tries to assist the human operator, boosting safety.
- **Car safety systems:** Automobile manufacturers are interested in making their cars aware of other cars or people close by and about to enter the path of vehicle. This would enable the car to alert the driver or to take evasive action by itself, so lowering accident rates.

For our thesis work, we focus on image-based object recognition, in which the goal is to determine what category or class an image belongs to, by using complete image means extracting the features from whole image. The model presented in this thesis is effectively used to achieve computationally efficient object recognition under a wide range of conditions as shown in Figure 1.11.

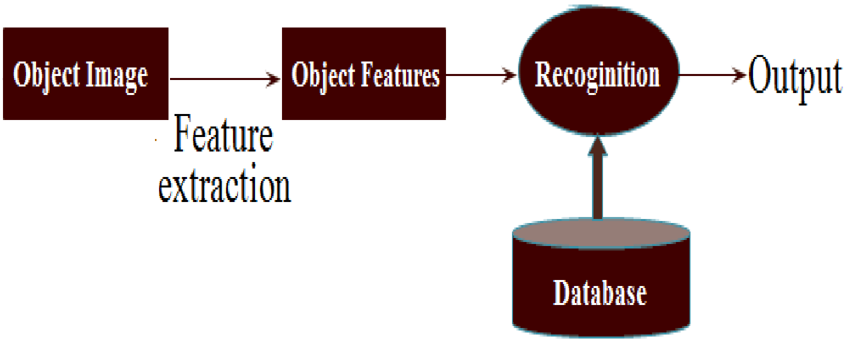


Figure 1.11: Proposed model for Object Recognition.

In our work we select commonly used method Gabor wavelet (GW) for feature extraction. Motivated by biological findings on the similarity of two-dimensional (2D) Gabor filters there has been increased interest in deploying Gabor filters in various computer vision applications and to texture analysis and image retrieval. The general functionality of the 2D Gabor filter family can be represented as a Gaussian function modulated by a complex sinusoidal signal [4]. In our work we use a bank of filters built from these Gabor functions for texture feature extraction. Before filtration, we normalize an image to remove the effects of sensor noise and gray level deformation. We adopt Otsu and GW to recognize the objects.

1.5 Thesis Outline

The organization of this thesis is as follows. Chapter 2 describes review of Gabor Wavelet feature extraction method. This chapter gives a brief discussion on Gabor Wavelet. As Gabor Wavelet method is used in our work in feature extraction for object recognition. Chapter 3 discusses review of various classifiers used in comparison. In this chapter we have given the process of these classifiers and the parameters set by us while comparison. Recognition System based on our proposed classifier is briefly described in Chapter 4. This chapter deals with the Fuzzy Inference Systems (FIS) for the classification of images, once they have been pre-processed through suitable algorithms. Chapter 5 presents experimental setup which includes feature extraction steps; in this Chapter we also describe the pre-processing method which is a thresholding method known as Otsu which increases the efficiency of our system. Chapter 6 will present the results and discussion; here we have shown the results of our system as well other systems with which we compared our system. Lastly in Chapter 7 the concluding remarks and future work are stated.

CHAPTER 2

REVIEW OF GABOR WAVELET

This chapter discusses the Gabor wavelets and how object images are represented by Gabor wavelet features. Firstly, Introduction about Gabor wavelets is given. Secondly, Gabor kernel is defined. Thirdly, the background of Gabor wavelets is given. Fourthly, the Gabor wavelet and its parameters are defined. Fifthly, Gabor wavelet transform is illustrated and the representatives of object images. And finally Gabor wavelet features are presented.

2.1 Introduction

Feature extraction and classifier learning are essential to the performance of a pattern recognition system. Features extracted should be as discriminative as possible. Classifiers should be robust enough to handle uncertainty in the data. Haar wavelets, Gaussian pyramid representation and Gabor wavelets are all multi-resolution based representations that have been used in object recognition/detection. Haar wavelets are an orthogonal basis that achieves a piece-wise constant approximation to the image [54]. In our work, we used Gabor wavelets (GWs) for feature extraction for object recognition. Our representation is unique in decomposing appearance along the dimensions of space, frequency, and orientation. Gabor features have been widely used in pattern recognition applications such as fingerprint recognition, character recognition, and texture segmentation etc. GWs use Gabor functions.

Gabor functions were first proposed in 1946 by Dennis Gabor [8] (a Hungarian physicist who is the most remarkable for inventing holography, for which he received the Nobel Prize in Physics in 1971) [49]. Gabor transform is the short-time Fourier transform, used to determine the sinusoidal frequency and phase content of a signal which changes with time. A complex Gabor filter is defined as the product of a Gaussian kernel times a complex sinusoid which is then transformed with a Fourier transform to derive the time-frequency analysis. The Gabor transform of a signal x(t) is defined by this formula [13] :

$$G_x(t,f) = \int_{-\infty}^{\infty} e^{-\pi(T-t)^2} e^{-j2\pi f^T} x(T)dT \qquad \textbf{..... (2.1)}$$

The Gabor representation was first introduced for the 1-D [8] then extending the representation to 2-D by Ebrahimi, et al. [50] show usefulness in image sequence coding where high compression ratios and modest image quality are required. Gabor filters with various scales and rotations are created and convolved with the signal, resulting in a so-called Gabor space. The Gabor space is very useful in image processing applications such as face tracking [31], face and object recognition [5,6], optical character recognition, iris recognition, fingerprint recognition, and texture analysis as well in neurophysical studies [30]. Various other advantages are:

- Gabor filter is a linear filter used for edge detection.
- Gabor filters (similar to the human visual system) used in time frequency and orientation analysis and helpful for extracting useful features from an image.
- Gabor maintains locality in the spatiotemporal-frequency domain, representing a promising candidates for use in future coding systems.
- Gabor filters are a popular tool for image analysis, and have found widespread use in computer vision.

The drawback to Gabor filtering in computer vision is high computational load required [13].

Gabor wavelets (GWs) with good characteristics of orientation and space-frequency localization are commonly used for extracting local features for various applications like object detection, recognition and tracking. The Gabor wavelet representation of an image is the convolution of the image with a family of Gabor wavelets [17]. Gabor wavelets detect the edge detector, face region and facial features regions. Gabor wavelets are widely used in image analysis and computer vision. The Gabor wavelets transform provides an effective way to analyze images and has been elaborated as a frame for understanding the orientation and spatial frequency selective properties of simple cortical neurons. They seem to be a good approximation to the sensitivity profiles of neurons found in visual cortex of higher vertebrates. The important advantages are infinite smoothness and exponential decay in frequency. The structures and functions of Gabor kernels are similar to the two-dimensional (2-D) receptive fields of the mammalian cortical simple cells [50]. To extract local features for pattern recognition the best method is to use Gabor wavelets for several reasons [12]:

• ***Biological motivation:*** J.G. Daugman discovered that simple cells in the visual cortex of mammalian brains can be modelled by Gabor functions.

• ***Mathematical motivation:*** A Gabor wavelet function consists of multi-resolution, multi-orientation properties as well as provides optimal space-frequency localization.

• ***Empirical motivation:*** Gabor wavelets yield distortion tolerance feature spaces for pattern recognition tasks.

The application of Gabor wavelets for object recognition has been pioneered by Lades et al.'s work since Dynamic Link Architecture (DLA) was proposed in 1993 [17]. In this system, faces and other objects are represented by a rectangular graph with local features extracted at the nodes using Gabor wavelets, resulting in Gabor jets. Rolf et al [5] presented a system for the recognition of human faces independent of hairstyle. Correspondence maps between an image and a model are established by coarse-fine matching in a Gabor pyramid and used for hierarchical recognition. Xing Wu [7] uses magnitude, phase, and frequency measures of the Gabor wavelet representation in an innovative flexible matching approach that can provide robust recognition. In this system the Gabor grid is a topology-preserving map that efficiently encodes both signal energy and structural information of an object in a sparse multi-resolution representation and subsamples the Gabor wavelet decomposition of an object model and is deformed to allow the indexed object model match with similar representation obtained using image data. Flexible matching between the model and the image minimizes a cost function based on local similarity and geometric distortion of the Gabor grid. Gabor wavelets have also been applied in global form for face recognition [12, 14-16]. These holistic methods normally extract Gabor features from the whole face image. Both linear and nonlinear subspace methods are applied thereafter for dimension reduction. A more detailed review on Gabor wavelet based face recognition methods can be found in [12]. Recently object features were extracted using Gabor wavelets and self-organizing maps in a hierarchical manner [6]. Object features are learned in an unsupervised way which is consistent with the feature learning process in the visual cortex. This algorithm presents a biologically inspired object recognition algorithm which is tolerant to two-dimensional (2D) affine transformations such as scaling and translation in the image plane and three-dimensional (3D) transformations of an object such as illumination changes and rotation in depth. The algorithm is analyzed for robustness. Gabor wavelets have the following properties that make them effective for pattern recognition:

- The shapes of Gabor wavelets are similar to the receptive fields of simple cells in the primary visual cortex.

- The Gabor function achieves the best time-frequency resolution for signal analysis.

Applications of Gabor wavelets for face processing are not only limited to recognition but in other applications such as for facial landmark location, tracking, head pose estimation and facial attribute classification etc [12]. The use of Gabor wavelet approach has several advantages such as robustness against facial expression, illumination, facial hair, glasses, image noise and invariance to some degree with respect to small changes in head pose. Further advantages include: More efficient and accurate compare to traditional approaches, saves neighbourhood relationships between pixels, easy to update, fast recognition and low computational cost [29].

2.2 Gabor Kernel

A two-dimensional Gabor filter is a linear filter whose kernel is similar to the two-dimensional (2-D) receptive profiles of the mammalian cortical simple cells in the primary visual cortex. They are localized in both space and frequency domains and have the shape of plane waves restricted by a Gaussian envelope function. Simple cells in the primary visual cortex have receptive fields (RFs) which are restricted to small regions of space and highly structured. Earlier examinations by Hubel and Wiesel leaded to a description of these cells as edge detectors. More recent examinations showed that the response behaviour of simple cells of cats corresponds to local measurements of frequencies. A 2-D Gabor kernel is a 2-D Gaussian function multiplied by a 2-D harmonic function. Generally, a harmonic function is a Fourier basis function. Especially, in a 2-D Gabor kernel it is a sinusoidally modulated function, in a form of complex exponential function. The Gaussian function varies in dilation and the harmonic function varies in rotation and frequency, so that a group of 2-D Gabor filters can be formed into 2-D Gabor wavelets. Gabor wavelet captures local structure corresponding to spatial frequency, i.e., scale, spatial localization, i.e., coordinate, and orientation selectivity information from a given signal, and the tuneable kernel size allows it to perform multi-resolution analysis. Among kinds of wavelet transforms, the Gabor wavelet transform has some impressive mathematical and biological properties and has been used frequently on researches of image processing. Gabor wavelets are used to extract texture information from object images [15,51,53].

2.3 Background Of Gabor Wavelets

2-D Gabor wavelet is widely adopted as a feature extraction approach in texture segmentation, iris recognition, face recognition, face expression recognition, and image retrieval. When Gabor filters are applied on computer vision or image processing task, one biggest problem is how to select the appropriate Gabor filters, i.e., the parameters. The parametric characterization has been studied extensively and different types of schemes have been emerged.

2-D Gabor filters transform different texture into detectable filter-output discontinuities at texture boundaries. Gabor filters is characterized as uniformly covering the spatial-frequency domain, and a filter selection scheme is presented based on minimal "energy" loss in reconstructed images from the filtered images. Teuner et al. choose parameters of Gabor filters based on the analysis of spectral feature contrasts obtained from iterations of pyramidal Gabor transforms. The work is benefit from no need for a prior knowledge of the texture image so that the segmentation processing is unsupervised. Since these parameterization solutions are all to choose optimal frequency or orientation of Gabor filter, an alternative wavelet scenario is proposed to bypass the optimization.

A 2-D Gabor wavelet model with multi-scale and multi-orientation is originally proposed by Daugman into biometric research. Daugman also has applied his 2-D Gabor wavelet model on human iris recognition. Liu and Wechsler applied the Enhanced Fisher linear discriminant Model (EFM) to an augmented Gabor feature vector derived from the Gabor wavelet representation of face images. Liu also presents a Gabor-based kernel Principal Component Analysis (PCA) method by integrating Gabor wavelet representation and the kernel PCA for recognition. Fan et al. combined Gabor wavelet and Null space-based Linear Discriminate Analysis (LDA) simultaneously on each orientation for generating feature vectors. The results show that as more Gabor wavelets are selected and applied, higher accuracy for recognition is achieved. In analysis of facial expression, the recognition is to analyze the relationship between the movements made by facial features, such as eyebrows, eyes and mouth. These facial features can be defined as point-based visual properties of facial expressions.

Hong et al. use Gabor wavelets of five frequencies and eight orientations to define a "big" General Face Knowledge (GFK) with 50 nodes 1 on a face, and a "small" 16-node GFK with three frequencies and four orientations. The method which fits these nodes with face image is the elastic graph matching proposed by Wiskott et al. in face recognition. Zhang et al. use 34 facial points for which a set of Gabor wavelet coefficients, the Gabor wavelets with three frequencies and six orientations have been utilised. Lyons et al. use a fiducial grid of 34 nodes and apply wavelets of five frequencies and six orientations [15,51]. Here we used four frequencies and four orientations for object recognition.

2.4 The Definition Of Gabor Wavelet

Gabor wavelets are introduced to image analysis due to their biological relevance and computational properties. Gabor transform is the short-time Fourier transform, used to determine the sinusoidal frequency and phase content of a signal which changes with time [13]. A Gabor wavelet (sometimes, called Gabor Kernel or Gabor Elemental Function or a complex Gabor filter) is defined as the product of a Gaussian kernel times a complex sinusoid which is then transformed with a Fourier transform to derive the time-frequency analysis and is defined as [15]:

$$\psi_{u,v}(z) = \frac{||k_{u,v}||^2}{\sigma^2} e^{-\frac{||k_{u,v}||^2 ||Z||^2}{2\sigma^2}} \left[e^{-ik_{u,v}Z} - e^{-\frac{\sigma^2}{2}} \right] \qquad \textbf{..... (2.2)}$$

where z = (x,y) indicates a point with horizontal and vertical coordinate of image obtained after Otsu thresholding. The operator $||.||$ denotes the norm operator. Parameters u and v defines the angular orientation and the spatial frequency of the Gabor kernel. In Equation 2.2 the spatial frequency modulates the size of the 2-D discrete Gabor kernel, so that v also determines scale of kernel. The parameter σ is the standard deviation of Gaussian window in the kernel. The wave vector $k_{u,v}$ is

$$k_{u,v} = k_v e^{i\emptyset_u} \qquad \textbf{….. (2.3)}$$

where $k_v = \frac{k_{max}}{f^v}$ and $\emptyset_u = \frac{u\pi}{4}$ if four different orientations have been chosen. k_{max} is the maximum frequency, and f is the spatial factor between kernels in the frequency domain.

2.4.1 Wavelets

A **wavelet** is a wave-like oscillation with amplitude that starts out at zero, increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one might see recorded by a seismograph or heart monitor. Generally, wavelets are purposefully crafted to have specific properties that make them useful for signal processing. Wavelets can be combined, using a "shift, multiply and sum" technique called convolution, with portions of an unknown signal to extract information from the unknown signal. Gabor kernels in Equation (2.2) are all self-similar differ only by a quadrature phase shift since they are generated from one kernel (from one mother wavelet) by dilation and rotation via the wave vector $k_{u,v}$. Each kernel is a product of a Gaussian envelope formulated in the equation

$$\frac{||k_{u,v}||^2}{\sigma^2} e^{-\frac{||k_{u,v}||^2 ||Z||^2}{2\sigma^2}} \qquad \textbf{….. (2.4)}$$

and a complex plane wave $e^{-ik_{u,v}Z}$. The complex wave determines the oscillatory part of the kernel. The term $e^{-\frac{\sigma^2}{2}}$ compensates for the Disparity Compensated (DC) value which makes the kernel DC-free [14]. The filter may have a large DC response. A popular approach to get a zero DC response is to subtract the output of a low-pass Gaussian filter. Subtracting the DC response, Gabor filters becomes insensitive to the overall level of illumination. DC-free is a wavelet terminology that ensures wavelets do not lose any generality such that there is no minimal energy loss when images are reconstructed by the wavelets. The effect of the DC term becomes negligible when the parameter σ, which determines the ratio of the Gaussian window width to wavelength, is a sufficiently large value.

2.4.2 Parameterization

The kernels exhibit desirable characteristics of spatial frequency, spatial locality, and orientation selectivity. In this thesis, 4 different scales and 4 orientations of Gabor wavelets are used, i.e. ν ∈ {0, . .,3}, and u ∈ {0, . .,3} in equation 2.2. The images adopted are with 40 × 40 size. The four orientations in radian are from 0 to 3π/4 with an interval of π/4. Gabor wavelets are modulated by a Gaussian envelope function with relative width σ = 2π, which is the standard deviation of Gaussian window. The maximum frequency is $k_{max} = \frac{\pi}{2}$, and the spatial factor is f = √2. These parameters are chosen according to previous findings [52].

2.4.3 Complex Gabor

Gabor kernel is a product of a Gaussian and a complex plane wave with real and imaginary parts, also called even and odd. An equation can be separated as real and imaginary parts like

$$e^{(x+iy)} = e^{x}(\cos(y) + i\sin(y)) \qquad \textbf{..... (2.5)}$$

Using the above equation the Equation 2.2 is separated into real and imaginary parts, so that the real part is [15]

$$\frac{{k_v}^2}{\sigma^2} e^{-\frac{{k_v}^2 ||Z||^2}{2\sigma^2}} \{\cos(k_v \cos(\phi_u)\, x + k_v \sin(\phi_u)\, y) - e^{-\frac{\sigma^2}{2}}\} \qquad \textbf{..... (2.6)}$$

and the imaginary part becomes

$$\frac{{k_v}^2}{\sigma^2} e^{-\frac{{k_v}^2 ||Z||^2}{2\sigma^2}} \sin(k_v \cos(\phi_u)\, x + k_v \sin(\phi_u)\, y) \qquad \textbf{..... (2.7)}$$

The real part and the imaginary part of 16 Gabor wavelets are shown in Figure 2.1 and Figure 2.2 respectively. These Gabor wavelets share 4 scales and 4 orientations. The orientations present from left to right and they are 0, π/4, π/2 and 3π/4. The scales from top to bottom are 0, 1, 2 and 3.

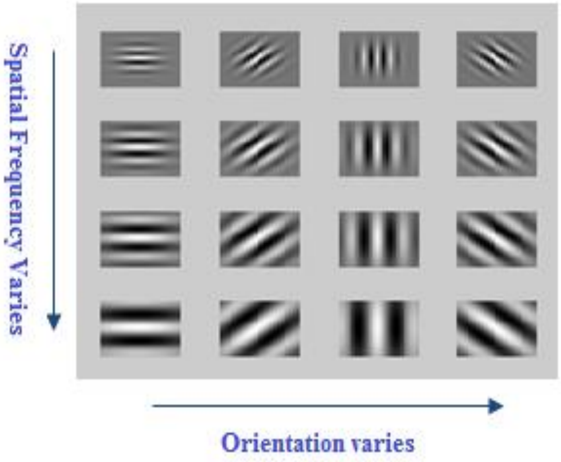


Figure 2.1: The real part of the 16 Gabor wavelets.

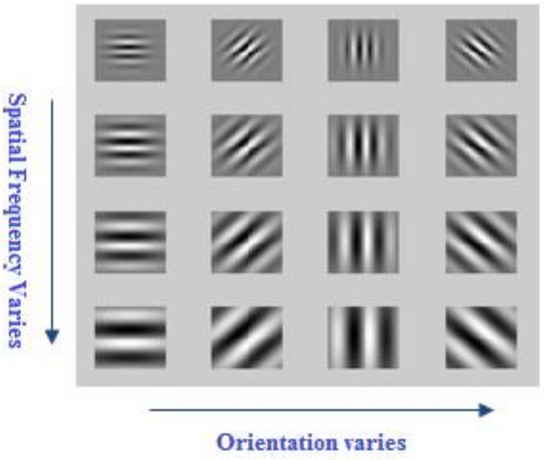


Figure 2.2: The imaginary part of the 16 Gabor wavelets.

2.5 Gabor Wavelet Transform

In computer vision, features stand as a piece of "interesting" information which is relevant for solving a specific vision application. In appearance based approaches, features refer to the result after a general neighbourhood operation applied on an image. The result contains local or global information which contributes to resolving a specific vision problem. An image can be represented by a group of features. A computer vision system works on these features rather than image pixels directly. Extraction of features is defined in terms of local neighbourhood operations. Gabor wavelet transform is the process to extract features which are relevant to object recognition. Since the selective schemes are available with a range of frequencies and orientation intervals, Gabor wavelets are ideal for object feature representation. Gabor functions are joint spatial and frequency domain measures, and are localized transformations in both domains. Gabor functions have many degrees of freedom that allow their spatial and spectral characteristics to be optimally adjusted to a specific visual requirement. Gabor wavelet filters have been used to solve a variety of image processing and computer vision problems.

This section illustrates the Gabor wavelet transform by convolution. First of all, the concept of convolution is given. Secondly, the 2-D discrete convolution is presented. Thirdly, the size of mask for convolution is determined. Finally, the magnitude response is used as extracted features [15,51,53].

2.5.1 Convolution

The Gabor wavelet representation of an image is the course of two-dimensional convolution of an image with a family of Gabor wavelet kernels defined by Equation 2.2. Two-dimensional (2-D) convolution is a specific type of local neighbourhood operation which belongs to the linear approach in image analysis. The 2-D convolution g of two-dimensional functions f and h is denoted by f * h. The function of 2-D convolution is expressed as g(x,y)= (f * h) (x,y). Similarly the convolution between image I(z) and a Gabor kernel $\psi_{u,v}(z)$ is defined as:

$$O_{u,v}(z) = I(z) * \psi_{u,v}(z) \qquad \textbf{..... (2.8)}$$

Let I(z), where z = (x, y) define the position in the image, the convolution of image I & a Gabor kernel $\psi_{u,v}(z)$ is shown where $*$ denotes the convolution operator, and $O_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at the orientation u and the spatial frequency v^2 .

2.5.2 2-D Discrete Convolution

The convolution is 2-D discrete convolution since the images and wavelets are both in two-dimension and discrete. In the 2-D discrete image domain, the linear filtering calculates the image pixel g(x, y) as a linear combination of the pixel value in a local neighbourhood of the pixel f(a, b). So the 2-D discrete convolution actually is a linear combination of the pixel in I(z) as a local neighbourhood of $\psi_{u,v}(z)$. Hence, $\psi_{u,v}(z)$ normally refers as Convolution Mask or Mask. The convolution mask is often used with an odd number of pixels in rows and columns, such as 3 × 3, 5 × 5 and so on.

2.5.3 The Size of Mask

In 2-D discrete convolution with images, the size of a Gabor filtering convolution mask is an important issue. A uniform and arbitrary size for all masks is not appropriate. If the size is small, the mask cannot involve all spatial extents of Gabor wavelets with high frequency. If the size is too large, the spatial extents of Gabor with high frequency only dominate a small portion of mask and rest portion lefts blank. Hence, during the convolution, computing on blank portion is useless. It should be large enough to show the nature of Gabor wavelets. However, it should not be too large, otherwise computation cost will be increased. For instance, in Figure 2.1, the size of the Gabor convolution masks is 50 × 50, and the Gabor wavelets with the lowest frequency are shown on the top row. The span of these Gabor masks only possesses a small part at the centre of the mask, and the rest of the area conveys no information. The size of a Gabor mask should be large enough to cover the shape of Gabor wavelet. The size of Gabor wavelet is determined by the spatial extent of the Gaussian envelope, which is then determined by the spatial frequency v and the deviation of Gaussian function σ in Equation 2.2. According to Dunn et al., the Gabor filter is truncated to six times the span of the Gaussian envelope. The span of Gaussian function is $\frac{\sigma}{||k_v||}$ and $k_v = \frac{k_{max}}{f^v}$ so that the Gabor mask is truncated to a width W

$$W = \frac{6\sigma}{||k_v||} + 1 = 6f^v \frac{\sigma}{k_{max}} + 1 \qquad \text{..... (2.9)}$$

Taking $k_{max} = \pi/2$, the factor $f = \sqrt{2}$ and the standard deviation of the Gaussian $\sigma = 2\pi$, the width is

$$W = 24 . 2^{\frac{v}{2}} + 1 \qquad \text{..... (2.10)}$$

For four different spatial frequencies ν ∈ {0, . . . , 3 }, the corresponding size of Gabor filtering masks are 19×19, 25×25, 35×35, and 49×49. Figure 2.3 shows with the 3π/4 orientation, the corresponding real masks with the different spatial frequencies.

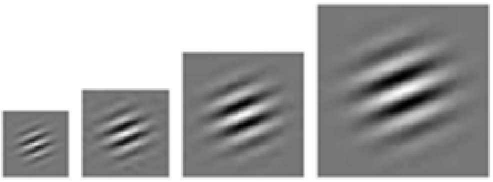


Figure 2.3: Gabor filtering convolution masks with the 4 different scales.

2.5.4 Magnitude Response

To get Gabor wavelet transform of an image, the 16 Gabor wavelets are convolved with the image. $O_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at the orientation μ and the spatial frequency v^2. Since Gabor wavelets are of the complex form, the convolution results contain the real response and imaginary response as follow:

$$O_{u,v}(z) = \Re\{ O_{u,v}(z) \} + i\,\Im\{ O_{u,v}(z)\} \qquad \text{..... (2.11)}$$

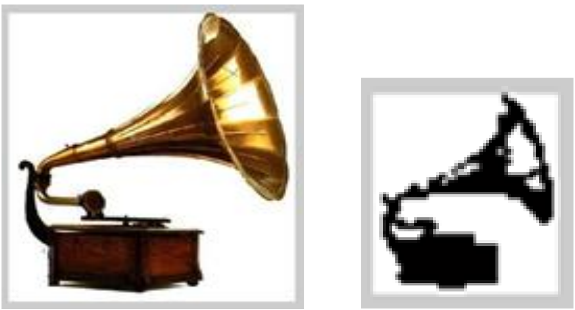
where $\Re\{ O_{u,v}(z) \}$ represents the real response and $\Im\{ O_{u,v}(z)\}$ represents the imaginary response. The real response of Gabor filtering is an image I(z) convolved with the real unit of Gabor kernel described as Equation 2.6. The real response of Gabor filtering is

$$\Re\{O_{u,v}(z)\} = \mathrm{I(z)} * \Re\{ \psi_{u,v}(z)\} \qquad \text{..... (2.12)}$$

The imaginary response is the image convolved with the imaginary part described as Equation 2.7. It is expressed as

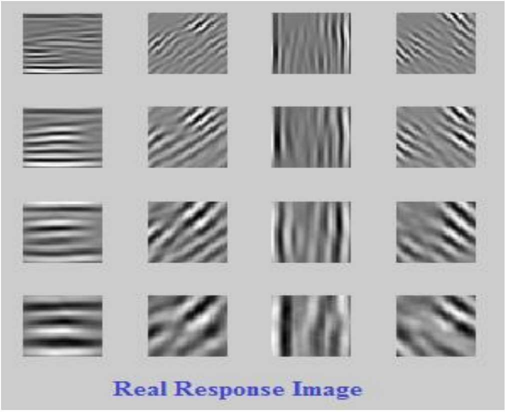
$$\Im\{O_{u,v}(z)\} = \mathrm{I(z)} * \Im\{ \psi_{u,v}(z)\} \qquad \text{..... (2.13)}$$

Given an object image shown in Figure 2.4(a) in the Caltech database [10-11] and corresponding thresholded image in Figure 2.4(b), then 16 Gabor real responses and imaginary responses are obtained corresponding to Figure 2.4(b) and displayed in Figure 2.5 and Figure 2.6 respectively. From Figures 2.5 and 2.6, it is hard to see that object rather than the stripes across the images.



(a) **(b)**

Figure 2.4: (a) The Object image from Caltech database and **(b)** its corresponding thresholded image.



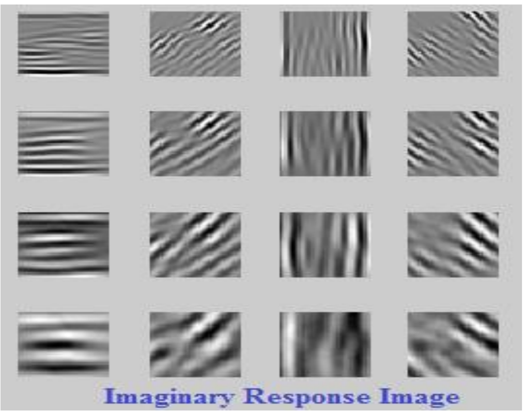


Figure 2.5: The 16 real response Images.

Figure 2.6: The 16 imaginary response Images.

Texture detection can be operated based on the magnitude of the output of the Gabor filtering. The magnitude response of Gabor filtering is widely used in texture detection. Since the object image contains various textures, the magnitude response of Gabor filtering will enhance the recognition of an object. The magnitude response $||O_{u,v}(z)||$ is the square root of the sum of the squared real response and imaginary response, and can be expressed as

$$||O_{u,v}(z)|| = \sqrt{\Re\{\, O_{u,v}(z)\,\}^2 + \Im\{\, O_{u,v}(z)\}^2} \qquad \textbf{..... (2.14)}$$

It can be seen that the magnitude response is modulus. These 16 Gabor magnitude responses are shown in Figure 2.7 relating to the image shown in Figure 2.4(b). The magnitude responses demonstrate local variance within low spatial-frequency and global variance within high spatial-frequency. In the top rows, more precise dissimilarity across the object are shown, while in the bottom rows, more high-level scale dissimilarity across the object are shown. The magnitude response of Gabor filtering is adopted to extract Gabor Wavelet Features.

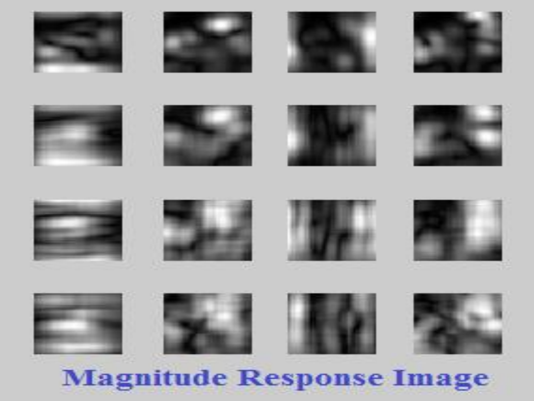


Figure 2.7: The 16 magnitude responses images

2.6 Gabor Wavelet Feature

In this thesis, the magnitude response $||O_{u,v}(z)||$ is used to represent the features. Therefore a Gabor wavelet feature j is configured by the three key parameters: the position z=(x, y), the orientation u, and the spatial frequency v. The value of a Gabor wavelet feature is the corresponding magnitude response of Gabor wavelet transform as

$$j(z, v, u) = ||O_{u,v}(z)|| \qquad \textbf{..... (2.14)}$$

Gabor wavelet features vary in orientation, frequency and position. There are four different orientations from 0 to 3π/4 with the interval π/4, and four different scales from 0 to 3. These 16 Gabor wavelets are applied on every position on an object image, so that the total number of Gabor wavelet features is determined by the number of orientations, the number of scales, and the resolution (size) of the image applied. Given an image I(z) with N × M pixels, the number of Gabor wavelet feature is N × M × 16. Hence, the representation is in a very high dimensional space. It is necessary to reduce it to a lower feature space. Gabor wavelets extract features from thresholded object images. For example, here is an image with 40 × 40 pixels and the total number of Gabor wavelet features is 40 × 40 × 4 × 4 = 25,600. If the number of Gabor wavelets is fixed, the only factor which changes the total number of features is the size of images. More precise details along different orientations can be reflected by Gabor wavelet transform [15].

CHAPTER 3
REVIEW OF CLASSIFIERS

In machine learning and statistics, **classification** is the problem of identifying which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. The individual observations are analyzed into a set of quantifiable properties, known as various explanatory variables, *features*, etc. An algorithm that implements classification, especially in a concrete implementation, is known as a **classifier**. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, which maps input data to a category. Classifications of objects are important areas in a variety of fields, such as pattern recognition, artificial intelligence and vision analysis. For recognition many different classifiers have been employed over the years like KNN, Neural Network (NN), GMM, HMM, SVM, LDA. In our work we proposed a new fuzzy classifier which uses GW features obtained from Otsu thresholded image and its performance analysis is carried out with respect to NN, KNN, LDA, SVM classifiers, which are explained as under and fuzzy classifier is explained in next chapter. In this chapter we also mentioned parameters selected for our work.

3.1 Nearest Neighbour Classifier

Euclid stated that the shortest distance between two points on a plane is a straight line and is known as Euclidean distance as shown in Equation 3.1 and is a non-parametric Classifier. In Euclidean distance metric difference of each feature of query and database image is squared which effectively increases the divergence between them.

3.1.1 Euclidean distance

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula.[55] By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. The Euclidean distance between points p and q is the length of the line segment connecting them ($\overline{\mathbf{pq}}$). In Cartesian coordinates, if p = (p_1, p_2,... , p_n) and q = (q_1, q_2,... , q_n) are two points in Euclidean n-space, then the distance from p to q, or from q to p is given by:

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}. \qquad \textbf{..... (3.1)}$$

The position of a point in a Euclidean n-space is a Euclidean vector. So, p and q are Euclidean vectors, starting from the origin of the space, and their tips indicate two points. The

Euclidean norm, or Euclidean length, or magnitude of a vector measures the length of the vector:

$$\|\mathbf{p}\| = \sqrt{p_1^2 + p_2^2 + \cdots + p_n^2} = \sqrt{\mathbf{p}\cdot\mathbf{p}}$$ **..... (3.2)**

where the last equation involves the dot product.

One dimension: In one dimension, the distance between two points on the real line is the absolute value of their numerical difference. Thus if x and y are two points on the real line, then the distance between them is given by:

$$\sqrt{(x-y)^2} = |x-y|.$$ **..... (3.3)**

Two dimensions: In the Euclidean plane, if p = (p1, p2) and q = (q1, q2) then the distance is given by

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2}.$$ **..... (3.4)**

Three dimensions: In three-dimensional Euclidean space, the distance is

$$d = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + (p_3-q_3)^2}.$$ **..... (3.5)**

N dimensions: In general, for an n-dimensional space, the distance is

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \ldots + (p_i-q_i)^2 + \ldots + (p_n-q_n)^2}.$$ **..... (3.6)**

3.1.2 Algorithm

Euclidean distance calculation is illustrated by an example in Figure 3.1. In this z1 and z2 are two known points of two different classes and x is a new point whose distance is calculated with respect to z1 and z2 using Equation 3.4 i.e. $d_E(x,z_1)$, $d_E(x,z_2)$ calculated respectively. If $d_E(x,z_1)$ is smaller then x is assigned to z_1 else z_2.

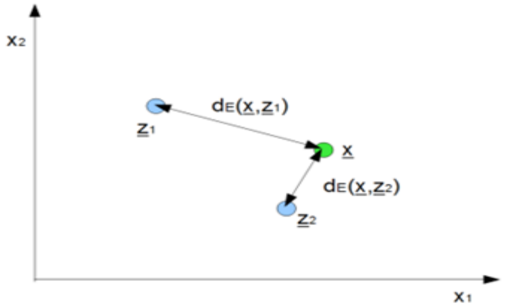


Figure 3.1: Minimum Euclidean distance Technique.

A most commonly used algorithm for image classification is the Euclidean classifier or Nearest Neighbour classifier. The classifier based on the Euclidean distance measure which is direct and simple. The mean class values are used as class centers to calculate pixel-center distances for use by the Euclidean distance rule. For major level classification of a homogeneous area this scheme is better. Its advantageous nature comes from the minimum time it takes to classify. In this algorithm each unknown pixel with feature vector x is classified by assigning it to the class whose mean vector (M) is closest to x [56]. With this method the clusters are approximated by N-dimensional spheres as shown in Figure 3.2.

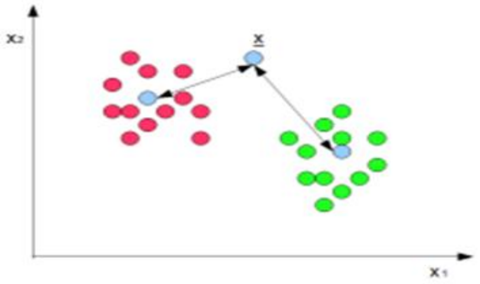


Figure 3.2: Example of Nearest Neighbour Classifier.

3.2 Naive Bayes Classifier

The Naive Bayes Classifier technique is based on the so-called Bayesian theorem with strong (naive) independence assumptions and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers. An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

3.2.1 Algorithm

The process of naive Bayesian classifier is illustrated by an example. To demonstrate the concept of Naïve Bayes Classification, consider the example illustrated in Figure. As indicated, the objects can be classified as either GREEN or RED. The task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently exiting objects. Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen. Thus, we can write:

$$Prior\ probability\ for\ GREEN \propto \frac{Number\ \ of\ GREEN\ objects}{Total\ number\ \ of\ objects} \qquad \textbf{..... (3.7)}$$

$$Prior\ probability\ for\ RED \propto \frac{Number\ \ of\ RED\ objects}{Total\ number\ \ of\ object\ s} \qquad \textbf{..... (3.8)}$$

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are:

$Prior\ probability\ for\ GREEN \propto \frac{40}{60}$ and $Prior\ probability\ for\ RED \propto \frac{20}{60}$

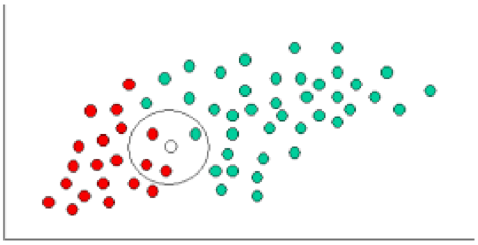


Figure 3.3: Example of Naive Bayes classification Technique

The prior probability has been formulated, and then a new object (WHITE circle) is classified as it arrives. Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, a circle around X is drawn which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then the number of points in the circle belonging to each class label is calculated. The likelihood:

$$Likelihood\ of\ X\ given\ GREEN \propto \frac{Num\,ber\ \ of\ GREEN\ in\ the\ vicinity\ \ of\ X}{Total\ number\ \ of\ GREEN\ cases} \qquad \textbf{..... (3.9)}$$

$$Likelihood\ of\ X\ given\ RED \propto \frac{Number\ \ of\ RED\ in\ the\ vicinity\ \ of\ X}{Tota\,l\ number\ \ of\ RED\ cases} \quad \textbf{..... (3.10)}$$

From the Figure 3.3, it is clear that Likelihood of X given GREEN is smaller than Likelihood of X given RED, since the circle encompasses 1 GREEN object and 3 RED ones. Thus:

$Probability\ of\ X\ given\ GREEN \propto \frac{1}{40}$ and $Probability\ of\ X\ given\ GREEN \propto \frac{3}{20}$

Although the prior probabilities indicate that X may belong to GREEN (given that there are twice as many GREEN compared to RED) the likelihood indicates otherwise; that the class membership of X is RED (given that there are more RED objects in the vicinity of X than GREEN). In the Bayesian analysis, the final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule.

$$Posterior\ probability\ of\ X\ being\ GREEN \propto$$

$$Prior\ probability\ of\ GREEN * Likelihood\ of\ X\ given\ GREEN = \frac{4}{6} * \frac{1}{40} = \frac{1}{60} \quad \textbf{.... (3.11)}$$

$$Posterior\ probability\ of\ X\ being\ RED \propto$$

$$Prior\ probability\ of\ RED * Likelihood\ of\ X\ given\ RED = \frac{2}{6} * \frac{3}{20} = \frac{1}{20} \quad \textbf{..... (3.12)}$$

Finally, we classify X as RED since its class membership achieves the largest posterior probability [57-58].

3.2.2 Parameter estimation

All model parameters (i.e., class priors and feature probability distributions) can be approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set. If one is dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution or to use binning to discretize the values. In general, the distribution method is a better choice if there is a small amount of training data, or if the precise distribution of the data is known. The discretization method tends to do better if there is a large amount of training data because it will learn to fit the distribution of the data. Since naive Bayes is typically used when a large amount of data is available (as more computationally expensive models can generally achieve better accuracy), the discretization method is generally preferred over the distribution method [58].

Choice of parameters in MATLAB: For Naive bayes classifier type of discriminant function is to be specified. Type is of:

- 'diaglinear' — Fits a multivariate normal density to each group, with a pooled estimate of covariance, but with a diagonal covariance matrix estimate.
- 'diagquadratic' — Fits multivariate normal densities with covariance estimates stratified by group, but with a diagonal covariance matrix estimate.

We also need to specify prior probabilities for the groups, prior is one of:

- A 'numeric vector' is the same length as the number of unique values in group (or the number of levels defined for group, if group is categorical). If group is numeric or categorical, the order of prior must correspond to the ordered values in group, or, if group contains strings, to the order of first occurrence of the values in group.
- A '1-by-1 structure' is useful if training is a subset a larger training set having fields:
    - prob — A numeric vector.
    - group — Of the same type as group, containing unique values indicating the groups to which the elements of probability correspond.
- The string 'empirical', indicating that group prior probabilities should be estimated from the group relative frequencies in training.

3.3 K-Nearest Neighbour Classifier

In pattern recognition, the K-nearest neighbour algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. It is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. K-nearest-neighbour classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine. In an unpublished US Air Force School of Aviation Medicine report in 1951, Fix and Hodges introduced a non-parametric method for pattern classification that has since become known the k-nearest neighbour rule. Later in 1967, some of the formal properties of the k-nearest-neighbour rule were worked out; for instance it was shown that for $k=1$ and $n\rightarrow\infty$ the k-nearest-neighbour classification error is bounded above by twice the Bayes error rate.

The *k*-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its *k* nearest neighbours (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of its nearest neighbour. The neighbours are taken from a set of objects for which the correct classification is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nearest neighbour algorithm is sensitive to the local structure of the data [59-60].

3.3.1 Algorithm

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the *k* training samples nearest to that query point. Usually Euclidean distance is used as the distance metric; another metric such as the overlap metric (or Hamming distance) can be used. To improve the accuracy of "k"-NN classification the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbour or Neighbourhood components analysis [60].

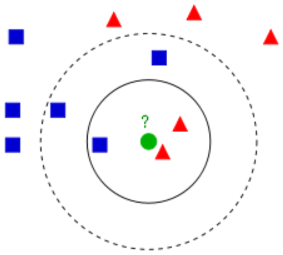


Figure 3.4: Example of *k*-NN classification Technique

Example of *k*-NN classification is shown in Figure 3.4. There are two classes 1st is triangle (shown in red color) and second is square (shown in blue color) and element shown in green circle is a test sample which is to be classified among one of these classes. *k*-NN classify on the basis of *k* value i.e. *k* nearest neighbours. If *k=3*, it is assigned to 1st class because there are 2 triangles and only 1 square inside the inner circle but if *k=5*, then it is it is assigned to the 2nd class because there are 3 squares and 2 triangles inside the outer circle.

3.3.2 Parameter selection

The best choice of *k* depends upon the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. The special case where the class is predicted to be the class of the closest training sample (i.e. when *k* =

1) is called the nearest neighbour algorithm. The accuracy of the *k*-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance [60]. For any classification to be carried out using the k-Nearest Neighbour classifier the distance metric has to be specified explicitly, which can be:

- 'euclidean' — Euclidean distance (default in MATLAB)
- 'cityblock' — Sum of absolute differences
- 'cosine' — One minus the cosine of the included angle between points (treated as vectors)
- 'correlation' — One minus the sample correlation between points (treated as sequences of values)
- 'hamming' — Percentage of bits that differ (suitable only for binary data)

Also, the rule that is used to decide how the sample has to be classified needs to be specified. It may be one of the following:

- 'nearest' — Majority rule with nearest point tie-break (In MATLAB, it is the default rule)
- 'random' — Majority rule with random point tie-break
- 'consensus' — Consensus rule

3.4 SVM Classifier

Originating from the hyperplane classifier proposed in [61], the support vector machine (SVM) has been greatly developed and widely applied in machine learning, classification and pattern recognition ever since. The original SVM algorithm was invented by Vladimir N. Vapnik and the current standard incarnation (soft margin) was proposed by Vapnik and Corinna Cortes in 1995. Support vector machine (SVM) [63] is the youngest part in statistical learning theory. It is a concept in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the input, making the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [64]. As a classifier, Support Vector Machines (SVM) are used to cluster data into two classes by finding the maximum marginal hyperplane that separates one class from the other by Boser et al. in 1992 [61]. The margin of the hyperplane, which is maximized, is defined by the distance between the hyperplane and the closest data points. The data points that lie on the boundary of the margin of the hyperplane are called the support vectors. SVMs have been used for multi-class classification problems [62]. In this method the input vectors are transferred to a high dimensional space where they are linearly separable. It uses different kind of kernel functions to perform the non-linear mapping into higher feature space.

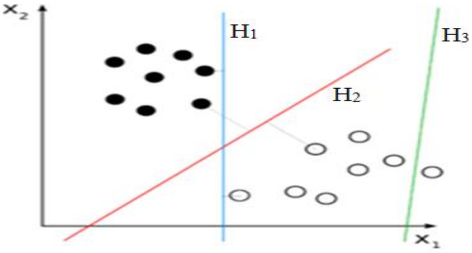


Figure 3.5: Example of SVM classification Technique.

3.4.1 Algorithm

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p − 1)-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perceptron of optimal stability. As shown in Figure 3.5, H_3 hyperplane in green doesn't separate the two classes. H_1 in blue separate two classes but with a small margin and H_2 in red also does with the maximum margin [64].

3.4.2 Nonlinear classification

The original optimal hyperplane algorithm proposed by Vapnik in 1963 was a linear classifier. However, in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be nonlinear in the original input space. If the kernel used is a Gaussian radial basis function, the corresponding feature space is a Hilbert space of infinite dimensions. Maximum margin classifiers are well regularized, so the infinite dimensions do not spoil the results. Other kernels are Polynomial (inhomogeneous), Polynomial (homogeneous), Hyperbolic tangent [64].

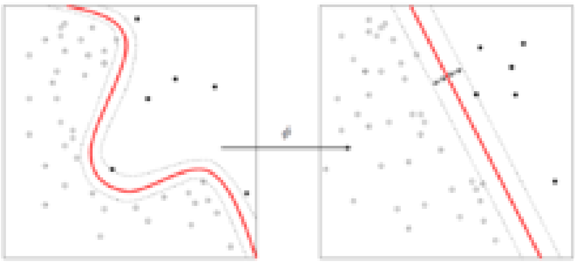


Figure 3.6: Example of Non-Linear and Linear SVM classification.

3.4.3 Multiclass SVM

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Common method for such reduction is done by building binary classifiers which distinguish between (i) one of the labels and the rest (*one-versus-all*) or (ii) between every pair of classes (*one-versus-one*). Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determines the instance classification. Crammer and Singer proposed a multiclass SVM method which casts the multiclass classification problem into a single optimization problem, rather than decomposing it into multiple binary classification problems [64].

3.4.4 Parameter selection

The effectiveness of SVM depends on the selection of kernel, the kernel's parameters, and soft margin parameter C. A common choice is a Gaussian kernel, which has a single parameter γ. Best combination of C and γ is often selected by a grid search with exponentially growing sequences of C and γ, for example, $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15}\}$; $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{1}, 2^{3}\}$. Typically, each combination of parameter choices is checked using cross validation, and the parameters with best cross-validation accuracy are picked. The final model, which is used for testing and for classifying new data, is then trained on the whole training set using the selected parameters [64]. In MARLAB for SVM Classifier Kernel Function Value has to be mention which is a function handle specifying the kernel function that maps the training data into kernel space. According to kernel function its value is mentioned. Choices are:

- 'linear' — Linear kernel or dot product (Default in MATLAB).
- 'quadratic' — Quadratic kernel.
- 'rbf' — Gaussian Radial Basis Function kernel with a default scaling factor, sigma, of 1.
- 'polynomial' — Polynomial kernel with a default order of 3.
- 'mlp' — Multilayer Perceptron kernel with default scale and bias parameters of [1, -1].
- '@functionname' — Handle to a kernel function specified using @and the functionname. For example, @kfun, or an anonymous function.

3.5 Neural Network Classifier

An artificial neural network (ANN), also called a simulated neural network (SNN) or commonly just neural network (NN) is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. In most cases a NN is an adaptive system that changes its structure based on external or internal information that flows through the network [49]. The earliest work in neural computing goes back to the 1940's when McCulloch and Pitts introduced the first neural network computing model. In the 1950's, Rosenblatt's work resulted in a two-layer network, the perceptron, which was capable of learning certain classifications by adjusting connection weights. Although the perceptron was successful in classifying certain patterns, it had a number of limitations. The perceptron was not able to solve the classic XOR (exclusive or) problem. Such limitations led to the decline of the field of neural networks. However, the perceptron had laid foundations for later work in neural computing. In the early 1980's, researchers showed renewed interest in neural networks. Recent work includes Boltzmann machines, Hopfield nets, competitive learning models, multilayer networks, and adaptive resonance theory models [65]. In more practical terms neural networks are nonlinear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

3.5.1 Algorithm

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. There are various types of neural networks. Explanation of all these types will not be possible due to constraint in space. But a specific type of neural network “The Multi-layer Perceptron (MLP)” will be dealt with in some detail since we are using a MLP for our object recognition system [49].

Multi-Layer Perceptron: This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function.

A neural network may be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The Figure 3.7 illustrates such a situation. Multi-layer networks use a variety of learning techniques, the most popular being back-propagation. Here the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network will usually converge to some state where the error of the calculations is small. In this case one says that the network has learned a certain target function. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network. To adjust weights properly one applies a general method for non-linear optimization task that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason back-propagation can only be applied on networks with differentiable activation functions.

The network performs well if it is able to classify the sample, which are very different from the training sample. This is especially important for cases where only very limited numbers of training samples are available. Computational learning theory is concerned with training classifiers on a limited amount of data. In the context of neural networks a simple heuristic, called early stopping, often ensures that the network will generalize well to examples not in the training set. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. Neural networks can also be trained to solve problems that are difficult for conventional computers or human beings. MATLAB provides four graphical tools for training neural networks to solve problems in function fitting, pattern recognition, clustering, and time series.

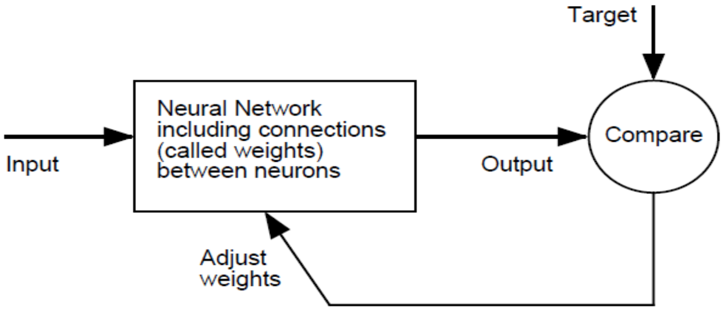

Target
Neural Network including connections (called weights) between neurons
Compare
Input
Output
Adjust weights

Figure 3.7: Neural Network Classifier

3.6 Parameter Selection for our Experiments

We have experimented with all the above classifiers. For experiments we need to set their parameters which are as follows: For NN classifier we have 30 neuron hidden layers. For KNN classifier number of nearest neighbours used in the classification is set to ‘1’, ‘Euclidean distance’ is chosen to calculate distance, rule with ‘nearest point’ is used to decide how to classify the sample in case of tie-break. For SVM classifier while training parameters such as kernel function which maps the training data into kernel space is chosen as ‘linear kernel or dot product’, ‘Sequential Minimal Optimization’ method is chosen to find the separating hyperplane, Box constraint is chosen as scalar value which is set as 1. After training a structure is returned with fields such as support vectors of size 48*25600, alpha of size 48*1, bias is 1.2829, kernel function, kernel function args, group names of size 150*1, support vector indices 48*1, scale data includes shift of size 25600*1 and scaleFactor of size 25600*1, figure handles which are used in classification process. For Naive bayes classifier discriminant function is set to ‘diaglinear’ type, ‘numeric vector of equal probabilities’ is used to specify prior probabilities for classes.

CHAPTER 4

REVIEW OF PROPOSED FUZZY CLASSIFIER

4.1 Fuzzy Logic

Fuzzy Logic is used to recognize objects, which was initiated in 1965, by Dr. Lotfi A. Zadeh [67] and it is based on the concept of "partial truth", i.e. truth values between "absolutely true" and "absolutely false". Basically, Fuzzy Logic is a multi-valued logic, which allows intermediate values to be defined between conventional evaluations like true/false, yes/no, high/low, etc. Fuzzy Logic provides a structure to model uncertainty, the human way of reasoning and the perception process. Fuzzy Logic is based on natural language and through a set of rules an inference system is built which is the basis of the fuzzy computation. Fuzzy logic has many advantages, firstly it is essential and applicable to many systems, moreover it is easy to understand and mostly flexible; finally it is able to model non linear functions of arbitrary complexity [66].

4.1.1 Membership Functions

An indirect way to represent a rule base is to define the set of the possible input values (in our case, membership function). The membership function is a graphical representation of the magnitude of participation of each input. It represents the degree of membership of a particular value to the specific fuzzy set. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. In simple words a membership function provides a measure of the degree of similarity of an element to a fuzzy set. It can take any form, but there are some common examples that appear in real applications. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different memberships functions associated with each input and output response [68].

Membership functions can

- either be chosen by the user arbitrarily, based on the user’s experience (MF chosen by two users could be different depending upon their experiences, perspectives, etc.).
- Or be designed using machine learning methods (e.g., artificial neural networks, genetic algorithms, etc).

For any set X, a membership function on X is any function from X to the real unit interval [0,1]. The membership function which represents a fuzzy set A is usually denoted by μ_A.

For an element x of X, the value $\mu_A(x)$ is called the membership degree of x in the fuzzy set A. The membership degree $\mu_A(x)$ quantifies the grade of membership of the element x to the fuzzy set A. The value 0 means that x is not a member of the fuzzy set; the value 1 means that x is fully a member of the fuzzy set. The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy set only partially. There are different shapes of membership functions [68]; triangular, trapezoidal, piecewise-linear, Gaussian, bell-shaped, etc. Some are explained here as under:

Triangular membership function: The equations corresponding to triangular MF ($\mu_A(x)$) of a fuzzy set A and is defined below, where a, b and c represent the *x* coordinates of the three vertices of $\mu_A(x)$ in a fuzzy set A. (a: lower boundary and c: upper boundary where membership degree is zero, b: the centre where membership degree is 1)

$$\mu_A(x)=\begin{Bmatrix} 0 & if\ x\le a \\ \dfrac{x-a}{b-a} & if\ a\le x\le b \\ \dfrac{c-x}{c-b} & if\ b\le x\le c \\ 0 & if\ x\ge c \end{Bmatrix}$$

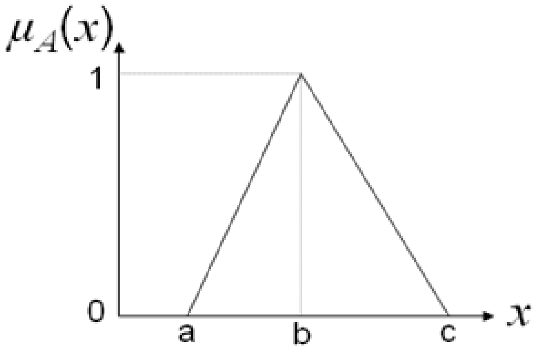
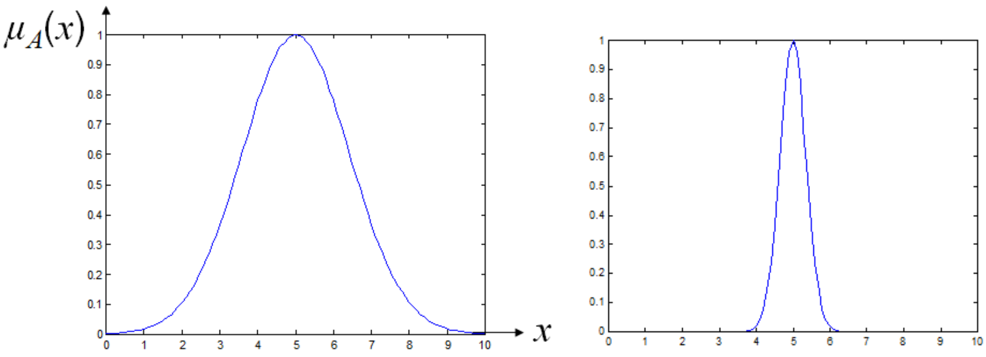


Figure 4.1: Triangular Membership function of a fuzzy set A.

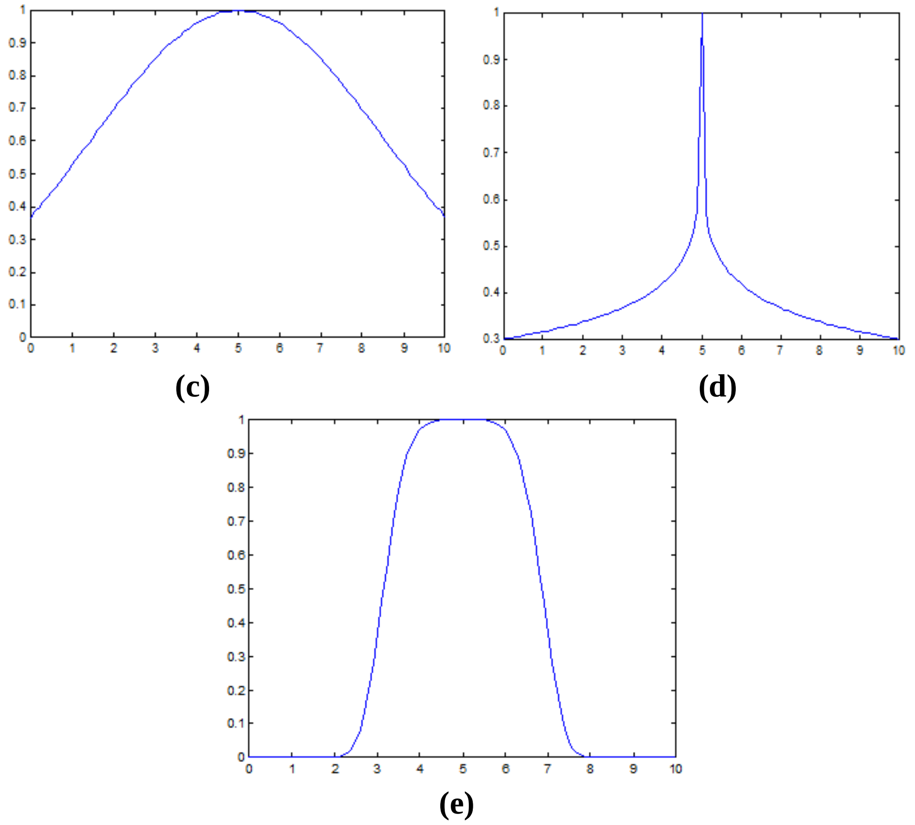
Gaussian membership function: The Gaussian curve is given by

$$\mu_A(x,c,s,m)=\exp\left[-\frac{1}{2}\left|\frac{x-c}{s}\right|^m\right] \qquad \textbf{..... (4.1)}$$

where *c* is the mean or refers to center in graph and σ is the variance used to vary width of curve, m: fuzzification factor (generally m=2).



(a) **(b)**



(c) (d)

(e)

Figure 4.2: Gaussian Membership function of a fuzzy set A with, **(a)** c=5, s=2, m=2; **(b)** c=5, s=0.5, m=2; **(c)** c=5, s=5, m=2; **(d)** c=5, s=2, m=0.2; **(e)** c=5, s=5, m=5

Generalized Gaussian membership function: We have used Generalized Gaussian membership function, whose curve is more flexible than Gaussian curve and is given by

$$\mu_s(x) = e^{-a(x-c)^b} \qquad \text{..... (4.2)}$$

We can relate equation 4.1 and 4.2, in 4.1 equation there is a constant term $\frac{1}{2S}$ similarly there is 'a' in 4.2 which is constant whose value we set to 1 and next there is m which is fuzzification factor whose value is taken as 2 in equation 4.1 while in 4.2 we have used a parameter variable 'b' whose value is taken as 1.66 explained in chapter 6.

4.2 Fuzzy Classification

Fuzzy classification is an application of fuzzy theory. Fuzzy classification is the process of grouping elements into a fuzzy set which allows its members to have different grades of membership (membership function) in the interval [0, 1]. In fuzzy classification an instance can belong to different classes with different membership degrees; conventionally the sum of the membership values of each single instance must be unitary. The main advantage of fuzzy classification based method includes its applicability for very complex processes.

4.2.1 Why fuzzy classifiers?

A classifier is an algorithm that assigns a class label to an object, based on the object description. It is also said that the classifier predicts the class label. The object description comes in the form of a vector containing values of the features (attributes) deemed to be relevant for the classification task. Typically, the classifier learns to predict class labels using a training algorithm and a training data set. When a training data set is not available, a classifier can be designed from prior knowledge and expertise. Once trained, the classifier is ready for operation on unseen objects [70].

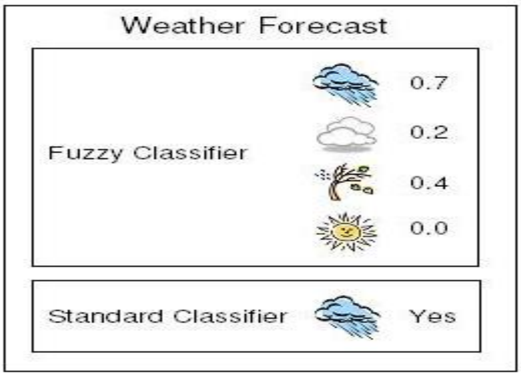


Figure 4.3: Fuzzy classifier produce soft class label.

Classification belongs to the general area of pattern recognition and machine learning.

- **Soft labelling:** The standard assumption in pattern recognition is that the classes are mutually exclusive. This may not be the case, as the example in Figure 4.3 shows. A standard classifier will assign a single crisp label (rain). A fuzzy classifier can assign degrees of membership (soft labels) in all four classes {rain, clouds, wind, sunshine}, accounting for the possibility of winds and cloudy weather throughout the day. A standard classifier can output posterior probabilities, and offer soft labelling too. However, a probability of, say, 0.2 for cloudy weather means that there is 20% chance that tomorrow will be cloudy. A probabilistic model would also assume that the four classes form a full group, i.e. snow, blizzards or thunderstorms must be subsumed by one of the existing four classes. Soft labelling is free from this assumption.
- **Interpretability:** Automatic classification in most challenging applications such as medical diagnosis has been sidelined due to ethical, political or legal reasons, and mostly due to the black box philosophy underpinning classical pattern recognition. Fuzzy classifiers are often designed to be transparent, i.e., steps and logic statements leading to the class prediction are traceable and comprehensible.

- **Limited data, available expertise:** Examples include predicting and classification of rare diseases, oil depositions, terrorist activities, natural disasters. Fuzzy classifiers can be built using expert opinion, data or both.

4.2.2 Fuzzy prototype-based classifiers

There are fuzzy classifier models inspired by the idea of "fuzzifying" conventional classifiers. A typical representative of this group is the K-nearest neighbor (KNN) classifier. In the classical KNN, the object x is labeled as the majority of its K nearest neighbors in a reference data set. The approximations of the posterior probabilities for the classes are crude, given by the proportion of neighbors out of k voting for the respective class. Fuzzy KNN uses the distances to the neighbors as well as their soft labels, if these are available. The reference set for this classifier does not have to be selected from the existing data. A set of relevant objects (prototypes) with crisp or soft labels can be constructed. The class membership of x is obtained through combining the similarities between x and the prototypes. Fuzzy prototype-based classifiers can be related to popular classifier models including Parzen classifier, learning vector quantization (LVQ) and radial basis functions (RBF) neural networks.

4.3 Proposed Fuzzy Classifier

The classifiers in each pool analyze the object regarding a certain aspect. Intuitively, a class is a set that is defined by a certain property, and all objects having that property are elements of that class. A classification is always made taking into account all the available classes, i.e., by means of a classification system. The process of classification evaluates for a given set of objects whether they fulfil the classification property, and consequentially are a member of the corresponding class. Therefore, a key concept in classification is the notion of partition, since it produces a structured family of classes. Each class is strongly related to each other, showing a specific structure (in a crisp context, for example, decision maker is being forced to choose one and only one class for each object). Fuzzy classification is the process of grouping elements into a fuzzy set [67] whose membership function is defined by the truth value of a fuzzy propositional function. Similarly in our work we proposed a new classifier using Generalised Gaussian membership function μ, for classification, given in equation 4.3.

The basic procedure to calculate degree of membership of test sample object image with a class is as follows: First degree of membership of the R component is calculated corresponding with a training object of a class. Similarly membership of G and B component is calculated corresponding to training object of a class. Then, the max of all the membership degrees is the membership degree of the object sample to that training object of a class. Now, this procedure is repeated and degree of membership of each object sample in the test dataset to all the training object of all classes in the training dataset is calculated. As, explained earlier, degree of membership will be a value between 0 and 1, that will indicate the membership of the frame to a particular class. Then, the object sample will be considered to belong to that training object of a class to which its degree of membership is the highest, and we find out class of that training object. Then test object sample will be assigned to that particular class. The step-by-step details are as given underneath:

4.3.1 Training and Test Dataset

In our real life we deal with color images so we have used color images for testing. But we have trained our network with Gray images means converting color images to Gray images then applying Otsu thresholding and then applying Gabor wavelet and obtaining features. The reason to use gray images is that size of single gray image's feature matrix is 25600×1 while that for RGB image is 25600×3. So this way we reduce the size of our training database. The feature matrix obtained for training is explained in detail in chapter 5. While for testing we obtain our feature vector in following way: Otsu thresholding method is applied on each component i.e. RGB components of color image separately. Then we apply Gabor wavelet transform as mentioned, on each component separately and getting a vector of size 25600×1 from each component. So now our test feature matrix for each image is of size 25600×3.

This way we obtain the feature matrix for training and testing image samples. So finally size of our Training dataset is 25600×150 (as we trained with 150 images) and that for Testing dataset is 25600×3×494 (as we tested 494 images).

4.3.2 Testing Phase

Recognition is the process of classifying objects to the trained object classes having the similar characteristics. The classifier, based on the features extracted, will classify object image sample into one of the classes which are used in training. For any sample to be classified, degree of membership of each object sample to each of the classes will be calculated. We have used Generalised Gaussian fuzzy membership function as was mentioned in Equation 4.2 we can relate it for our work to calculate degree of membership, which is as follows:

$$\mu_{s,i}(x_i) = e^{-a(x_i - u_{s,i})^b} \qquad \textbf{..... (4.3)}$$

where a and b represents the variables whose values obtained experimentally explained later-on in chapter 6. It was observed that fixing their values to a = 1 (by popular choice) and b = 1.66, we obtain best results. x_i denotes i^{th} feature of test image and $u_{s,i}$ denotes corresponding i^{th} feature of s^{th} training image sample respectively, so i denotes the features with i =1 to 25600 and s denotes sample object images with s = 1 to 150. From equation 4.3 we calculate degree of membership of a test image sample corresponding to an image in training dataset.

So Generalised Gaussian fuzzy membership function for our system can be written as

$$\mu_{s,i}(x_i) = e^{-(x_i - u_{s,i})^{1.66}} \qquad \textbf{..... (4.4)}$$

for each i value where i represents the features with a range from 1 to 25600 (here we have used actual difference instead of absolute values after difference at exponential power). The difference at power is done for each feature of test object image and corresponding feature number of every input image. Summation of all the features is used which takes algebraic sum of all the features. So summation for generalization of logical conjunction, for fuzzy logics is used as follows:

$$\sum_{i=1}^{25600} \mu_{s,i}(x_i) = \mu_{s,1}(x_1) + \mu_{s,2}(x_2) + \ldots\ldots + \mu_{s,25600}(x_{25600}) \qquad \textbf{..... (4.5)}$$

Now, testing is done on a sample by sample basis. Repeat all steps for every test samples:

Step 1: Each test sample is comprised of 3 channels (RGB). Using each feature value in every channel of test sample, the degree of membership to each training object image using its corresponding feature values is calculated by equation 4.4 & 4.5 which are as follows:

$$\mu_{s,j} = \sum_{i=1}^{25600} \mu_{s,i,j}\left(x_{i,j}\right) \qquad \textbf{..... (4.6)}$$

where j= R, G, B. For each j value where j represents three channels and i represents the features with i = 1 to 25600. The equations 4.4 & 4.5 are used for one dimension and for RGB we calculate 3 different values called as $\mu_{s,R}$, $\mu_{s,G}$, $\mu_{s,B}$ using equation 4.6. Means we took an image from a test dataset and we calculate degree of membership for R, G and B component corresponding to an object image from training dataset.

Step 2: Then we select that channel's membership value corresponding to each object image sample from training dataset whose degree of membership value, μ_s, is the maximum amongst training object image samples shown in equation 4.7. Means that channel from R, G and B is selected whose degree of membership corresponding to an image from training dataset is maximum.

$$\mu_s = \max_{j=R,G,B}\left(\mu_{s,j}\right) \qquad \textbf{..... (4.7)}$$

Step 3: Similarly we calculate degree of membership of test sample object image corresponding to each image in training dataset using step 1 and 2 explained above. Then assign the test sample to that training object sample which has maximum degree of membership value, µ amongst all as shown in equation 4.8. Means first we calculate degree of membership of selected components corresponding to each object image from training set. Then from those selected components we again find out maximum degree of membership is obtained with which training object image sample.

$$\square = \max_{1 \leq s \leq 150} (\mu_s) \qquad \text{..... } \mathbf{(4.8)}$$

Step 4: Finally, each test sample is classified to the most appropriate object class. As in last step 3 test sample was assigned to training object sample, in this step we check to which object class that training object sample belongs to (means from 1 to 10):

$$Allocated\ Class\ For\ Test\ Sample = [\max_{1 \leq s \leq 100} (\mu_s)] \qquad \text{..... } \mathbf{(4.9)}$$

This way all these steps are applied on each test sample for recognition and finally we get the class to which each test sample is classified.

So from above discussion it is clear that because of fuzzy logic we are able to classify our color images using gray image prototypes. This makes the proposed algorithm independent of the actual color combination of the test image. The results are better than actually classifying Gray to Gray or Color to Color as observed in the case of nearest neighbour classifier. The execution time however is comparable to the nearest neighbour classifier due to simple computations. As we are having the testing images in RGB form i.e. 3 dimensional format and our training dataset is in Gray form i.e. 1-dimensional format so here distance metrics for classification does not work well as can be seen from the results from Table 6.3 for second last column where as our fuzzy logic is able to do such classification with high efficiency can be seen in last column of Table 6.3. So here our Fuzzy logic will dominate and we can say that here in our work fuzziness is present. That is the reason to choose the fuzzy logic instead of distance metric to classify 3-D images from 1-D training sets in our work.

CHAPTER 5

EXPERIMENTAL SETUP

Each dataset was split into a test dataset and a training dataset. The training dataset was used in learning an object model and the test dataset was reserved for testing the model. The Caltech 101 dataset [11] is used for training and testing, and images from the database are processed for the experimental purpose. The experiments have shown interesting and positive results in object recognition. The experiment includes two steps: segmentation and feature extraction. Segmenting each of the datasets consisted primarily of extracting features, where we are enhancing the object features in an image removing noise to some extent. After that for each object class model, features are extracted from the training data. On the line of work done by [35- 37] we adopt Otsu (as segmentation step) and GW (as feature extraction step) to recognize the objects. We first apply Otsu thresholding and then Gabor Wavelet. So the experiment includes two stages for feature extraction: first using Otsu thresholding then applying Gabor filter. Following this step, the learning procedure is then used to train our model so that objects can be recognized efficiently.

So in this chapter first we describe our dataset. Secondly we describe segmentation using Otsu method and finally feature extraction using Gabor wavelet.

5.1 Dataset

In this section, we present the images database [10-11] which is used in our research. The experimental work is performed on object samples taken from the Caltech dataset. The Caltech 101 dataset consists of images of various objects. Caltech 101 is a dataset of digital images created in September, 2003, compiled by Fei-Fei Li, Marco Andreetto, Marc Aurelio Ranzato and Pietro Perona at the California Institute of Technology. It contains a total of 9146 images of objects belonging to 101 categories (including faces, watches, ants, pianos, chairs, guitars, etc). Most categories have about 50 images with the size of 300 * 200 pixels. It is intended to facilitate Computer Vision research and techniques. It is most applicable to techniques interested in recognition, classification, and categorization.

So in our research work we have included 10 different objects classes from Caltech 101 dataset selecting all the images present in the selected classes. We have included objects classes such as Butterfly, Ketch, Garfield, Gramophone, Electric Guitar, Hedgehog, Mandolin, Menorah, Panda, and Pyramid and their corresponding images can be seen in Figure 5.1. Fifteen exemplars were chosen randomly from each of the 10 object classes for training and remaining images are chosen for testing, yielding a total 644 exemplars. The no of images originally contained in Caltech 101 dataset in each class are 81 images in Butterfly class, 34 in Ketch, 51 in Garfield, 75 in Gramophone, 54 in Electric Guitar, 114 in Hedgehog, 43 in Mandolin, 87 in Menorah, 38 in Panda, and 57 in Pyramid. Therefore it consists of a total of 644 images. Out of these 150 images are used for training and rest 494 images are used for testing. Figure 5.1 shows the images used for training.

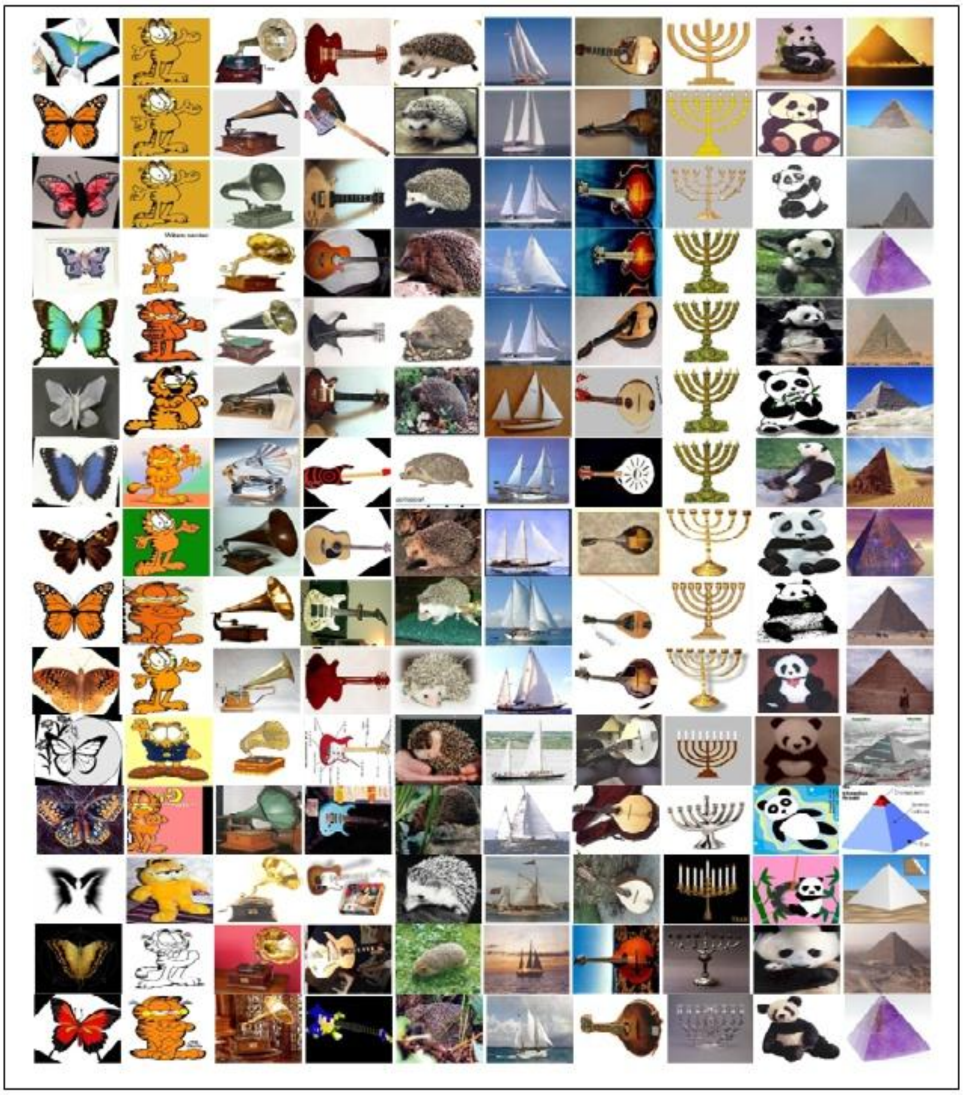


Figure 5.1: Images of Objects used in training our system for object recognition.

5.2 Segmentation

Segmentation is a process of grouping image pixels into units that are homogeneous with respect to one or more characteristics. Color image segmentation is useful in many applications. From the segmentation results, it is possible to identify regions of interest and objects in the scene, which is very beneficial to the subsequent image analysis [72]. Image segmentation is the key step from image processing to image analysis. Its purpose is to separate target and background and provide a basis for the follow-up actions of computer vision. Image segmentation methods include thresholding, edge detection, and so on. It is the first step of automatic target recognition.

There are primarily four types of segmentation techniques: thresholding, boundary based, region based and hybrid technique. ***Thresholding techniques*** [71] are fundamental for image segmentation. It is often realistic to assume that each pixel is subject to the mixture of several normal distributions, and many methods of determining suitable thresholds were proposed from a histogram of gray level intensity of pixels on the above assumption. Otsu method is one of the traditional threshold selection methods, which is based on variance and intensity. ***Boundary-based methods*** assume that the pixel properties, such as intensity, color and texture, should change abruptly between different regions. ***Region based methods*** assume that neighbouring pixels within the same regions should have similar values. ***Hybrid methods*** tend to combine boundary detection and region growing together to achieve better segmentation [72]. In our work we selected thresholding method for segmentation.

5.2.1 Thresholding

Analysis of binary images is an important part of computer vision. In most practical situations, gray-level images are converted to binary images after thresholding. There are innumerable techniques for binarizing gray-scale images [71]. One of the simplest techniques is to find a threshold based on the histogram of the image. Image pixels with gray levels above the threshold are classified as object pixels and the rest are classified as background pixels. Ideally, the histogram of the image to be binarized will be bimodal, and the threshold can be chosen in the valley between the two peaks. However, when the peaks are not pronounced and the histogram is not smooth, locating the valley could be difficult. For this reason, researchers have used several criteria to determine the threshold.

Image thresholding is the process of classifying image gray values into two or more classes. The gray level histogram is usually the starting point for image classification. Automatic thresholding is an important technique in image segmentation and machine vision applications. The basic idea of automatic thresholding is to automatically select an optimal gray-level threshold value for separating objects of interest in an image from the background based on their gray-level distribution. Thresholding technique has been widely used in the industry for automated visual inspection of defects, for blueprint images based on geometrical features. This technique is often referred to as contrast sensing in the machine vision industry. Because of its wide applicability to other areas of image processing and applications, there is a considerable body of work on automatic thresholding. In-depth surveys of various thresholding methods are given by Sahoo et al., Sezgin and Sankur, and Lee et al. More recent studies on this subject can be found in Sauvola and Pietikainen, and Yen et al. Some thresholding methods (e.g., valley seeking methods) use shape of an image gray-level histogram suffers many difficulties. To utilize the information in the histogram, direct use of the shape of the histogram can be avoided by using some criteria functions derived from the histogram data. Entropy may be used as a criteria function of thresholding. Thresholding methods based on the entropy function do not always give a good solution. Sometimes results obtained by the entropic thresholding methods are found to be biased [74-75]. Automatic thresholding techniques can be roughly categorized as global thresholding and local thresholding [76].

Global Thresholding Methods: Global thresholding calculates a single threshold value for the whole image from the histogram of the entire image. Pixels with a gray level under the threshold level are labelled as print; pixels with a gray level above the threshold level are labelled as background. The process can be described as follows:

$$b(x,y) = \begin{cases} 1 & \text{if} \quad f(x,y) \le T \\ 0 & \text{otherwise} \end{cases} \qquad \text{..... (5.1)}$$

Among many global thresholding methods, Otsu's method appears to be the most efficient. Otsu's method [9] is based on the analysis of the gray level histogram of the whole image and chooses an optimal threshold according to the discriminant theory. We implemented Otsu's method is used for thresholding in our experiments.

Local Thresholding Methods: Local thresholding uses localized gray-level information to choose multiple threshold values; each is optimized for a small region in the image. There are many local thresholding methods published. Trier and Jain compared 11 locally adaptive binarization methods and Nilblack's method gave the best results. Nilblack's method is based on the calculation of the local mean of and local standard deviation. The threshold is decided by the following formula:

$$\mathrm{T}(\mathrm{x},\mathrm{y}) = \mathrm{m}(\mathrm{x},\mathrm{y}) + \mathrm{k} * \mathrm{s}(\mathrm{x},\mathrm{y}) \qquad \text{..... (5.2)}$$

where m(x, y) and s(x, y) are the average of a local area and standard deviation values, respectively. The size of the neighbourhood should be small enough to preserve local details, but at the same time large enough to suppress noise. The value of k is used to adjust how much of the total print object boundary is taken as a part of the given object.

Global thresholding is simpler and easier to implement but its result relies on good (uniform) illumination. Local thresholding methods can deal with non-uniform illumination but they are slow. For visual inspection applications, where non-uniform illumination is usually not an issue, global thresholding is commonly used for its simplicity and speed.

5.2.2 Otsu Thresholding

Thresholding techniques are important for image segmentation which helps in extracting objects from their background. It is a simple shape extraction technique. It is used when the brightness of the shape is known; in fact pixels forming the shape can be detected categorizing pixels according to a fixed intensity threshold. The main advantage lies in its simplicity and in the fact that it requires a low computational effort but this approach is sensitive to illumination change and this is a considerable limit. So to increase the performance of our system we use Otsu thresholding method which is used to localize the objects more efficiently. Otsu thresholding came into picture in 1979 [9], where a method was presented to automatically select a threshold from a gray level histogram with the viewpoint of discriminant analysis. Otsu's method is a very popular global automatic thresholding technique. The Otsu thresholding method has been widely used in image thresholding in a wide range of applications, such as medical image processing, noise reduction for human action recognition [32], adaptive progressive thresholding to segment lumen regions from endoscopic images[33], document segmentation, pre-processing of a neural-network classifier for hardwood log inspection using CT images, low cost in-process gauging system in removing illumination dependencies well, real-time segmentation of images with complex background environment and segmentation of moving lips for speech recognition. These applications demand real-time performance and a hardware implementation is essential to increase the computational efficiency of the Otsu's procedure [77]. Otsu was widely used for object detection in [41-46], and in other applications like binarization of blueprint images [If], binary Logarithmic Conversion Unit (LCU), defect detection [75], edge detection [36] and Color Image Segmentation [34].

Otsu thresholding, which features a good performance, is one of the main image threshold segmentation methods. Otsu method is a popular method in computer vision and image processing, used to automatically calculate the thresholding level with which a gray level image is reduced to a binary image. The algorithm assumes that the image to be thresholded contains two classes of pixels (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal. Sahoo et al.'s study concluded that the Otsu thresholding method was one of the best threshold selection methods for general real world images with respect to uniformity and shape measures. The Otsu thresholding method is based on a very simple idea: finding the threshold that maximises the between-class variance. The Otsu method is optimal for thresholding large objects from the background. It indicates that the Otsu method is fine for thresholding a histogram with bimodal or multi-media distribution. In gray scale image thresholding, the task is to separate the foreground and background. The foreground may be objects needing to be tracked or processed, and the background is anything other than the specified objects. For example, in Optical Character Recognition (OCR), binary operation is required for document segmentation. The thresholding method is to separate some pixels belonging to characters and other pixels belonging to white-space. In gray scale images, pixel values range from 0 to 255 in 256 integers. Each integer is taken as a possible threshold to segment the foreground and the background. For each possible threshold, the between-class variance is computed. From numerous possible thresholds, the optimal threshold is the one which has the maximum variance [15]. The detailed algorithm for the Otsu thresholding method can be found in [9][75][78][79] as explained next.

Two-Dimensional Otsu Method: An image can be represented by a 2D gray-level intensity function f(x, y). The value of f(x, y) is the gray-level, ranging from 0 to L-1, where L is the number of distinct gray-levels. Let the number of pixels with gray-level i be n_i, and n be the total number of pixels in a given image, the probability of occurrence of gray-level i is defined as [9][75][78][79]:

$$P_i = \frac{n_i}{n} \qquad \textbf{..... (5.3)}$$

The average gray-level of the entire image is computed as:

$$\mu_T = \sum_{i=0}^{L-1} ip_i \qquad \textbf{..... (5.4)}$$

In the case of single thresholding, the pixels of an image are divided into two classes C_1 = {0, 1, …, t} and C_2 = {t+1, t+2,…, L-1}, t is the threshold value. C_1 and C_2 are normally corresponding to the objects of interest and the background. The probabilities of the two classes are:

$$w_1(t) = \sum_{i=0}^{t} p_i \; and, w_2(t) = \sum_{i=t+1}^{L-1} p_i \qquad \textbf{..... (5.5)}$$

Thus, the means of the two classes can be computed as

$$\mu_1(t) = \sum_{i=0}^{t} \frac{ip_i}{w_1(t)} \; and, \mu_2(t) = \sum_{i=t+1}^{L-1} \frac{ip_i}{w_2(t)} \qquad \textbf{..... (5.6)}$$

Using discriminant analysis, Otsu [4] showed that the optimal threshold t* can be determined by maximizing the between-class variance; that is:

$$t^* = Arg \; \max_{0 \le t \le L} \{\sigma_B^2(t)\} \qquad \textbf{..... (5.7)}$$

Where the between-class variance σ_B is defined as:

$$\sigma_B^2(t) = w_1(t)(\mu_1(t) - \mu_T)^2 + w_2(t)(\mu_2(t) - \mu_T)^2 \qquad \textbf{..... (5.8)}$$

An equivalent, but simpler formulation for the Otsu method is given in Liao et al. [2]. The simplified formula for obtaining optimal threshold t* is as follows

$$t^* = Arg\ \max_{0 \leq t \leq L}\{w_1(t){\mu_1}^2(t) + w_2(t){\mu_2}^2(t)\} \qquad \textbf{..... (5.9)}$$

The Otsu method described here can be easily extended to multilevel thresholding of an image [2,4]. For M-1 thresholds, which divide the image pixels into M classes, C_1~C_M, the optimal thresholds $\{t_1^*, t_2^*, \ldots, t_{M-1}^*\}$ are chosen by maximizing the between-class variance as follows:

$$\{t_1, t_2 \ldots t_{M-1}\} = Arg\,\{\,\max_{0 \leq t \leq L-1}\{\sigma_B^2(t_1, t_2 \ldots t_{M-1})\}\,\} \qquad \textbf{..... (5.10)}$$

which can also be written as:

$$\{t_1^*, t_2^*, \ldots, t_{M-1}^*\} = \ Arg\left\{\,\max_{0 \leq t_1 < \cdots < t_{M-1} \leq L}\{\textstyle\sum_{k=1}^{M} w_k u_k^2\}\,\right\} \qquad \textbf{..... (5.11)}$$

The Otsu method works well when the images to be thresholded have clear peaks and valleys. In other words, it works for images that their histograms show clear bimodal or multimodal distributions. Figure 5.2 & 5.3 shows the results obtained from 2D Otsu thresholding method.

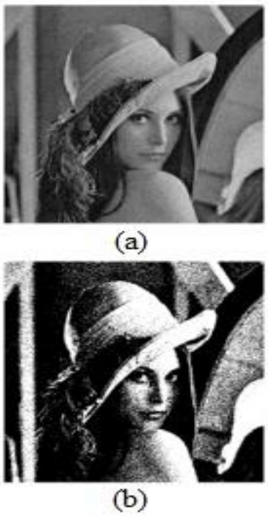

(a)
(b)

Figure 5.2: (a) Original Image **(b)** 2-D Otsu's method

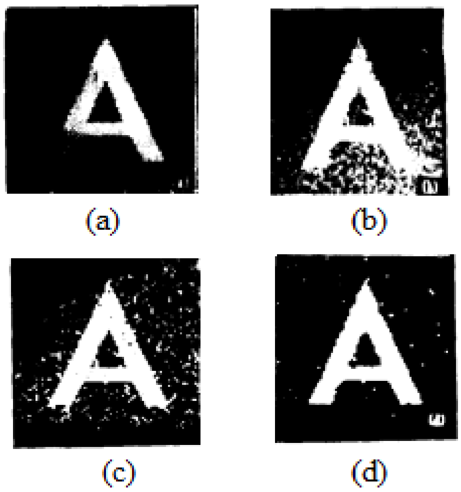

(a) (b)
(c) (d)

Figure 5.3: (a) Pure Image. **(b)** Image of 5.3(a) corrupted by N (0.100) noise. **(c)** Image after thresholding using 1D Otsu method. **(d)** Image after thresholding using 2D Otsu method

5.3 Feature Extraction Method

Feature extraction tends to simplify the amount of property required to represent a large set of data correctly. A feature can be defined as a function concerning measurements which represent a property of a considered object.

Feature extraction is done using Gabor filter on thresholded image. While there are many applications for recognition, but a lot of difficulties exists as were described earlier. The Gabor filter masks can be considered as orientation and scale tunable edge and line detectors. The statistics of these micro features in a given region can be used to characterize the underlying texture information. Gabor wavelet based texture is robust to orientation and illumination change, It is a powerful tool to extract texture features. Gabor functions are Gaussians modulated by complex sinusoids.

Gabor Wavelet and Otsu thresholding together came into existence in 2000 where Gabor wavelets was used to reduce the redundancy in the wavelet-based representation and Otsu's method of thresholding was used to reconstruct the magnitude and phase of the directional components of the image [35], also overcomes the shortcomings of CANNY algorithm with the ability of automatic edge detection from multi-scale and multi-dimensional [36], also used in analysis of retinal blood vessels which is extremely important for diagnosis and treatment of many diseases. So vessel segmentation is one of the most critical step for detection and treatment of diseases. It uses GW and Otsu for vessel segmentation in retinal images [37]. In all these researches first GW is applied then thresholded by Otsu method.

Feature extraction is the first and crucial step for obtaining the object information. To extract the features from images using Otsu and GW following steps are carried out (shown in Figure 5.4):

Step 1: First converting an RGB input image to Gray-scale and reducing its size to 40×40. This step is done to reduce the size of feature matrix obtained for training.

Step 2: Image segmentation plays an important role in image analysis and computer vision system. Among all segmentation techniques, the automatic thresholding methods are widely used because of their advantages of simple implement and time saving. Otsu method is one of thresholding methods and frequently used in various fields. Two-dimensional (2D) Otsu method behaves well in segmenting images of low signal-to-noise ratio than one-dimensional (1D). So Otsu threshold method is applied as a pre-processing step in order to remove noise and binarize the image.

Step 3: Gabor Wavelet filter is created and the parameters for Gabor wavelet are set as Gabor kernel size is taken as 24*24, orientations 0, π/4, π/2, 3π/4 and scales 0,1,2,3. The Kernel size is not taken smaller or larger than image size so that appropriate information can be determined.

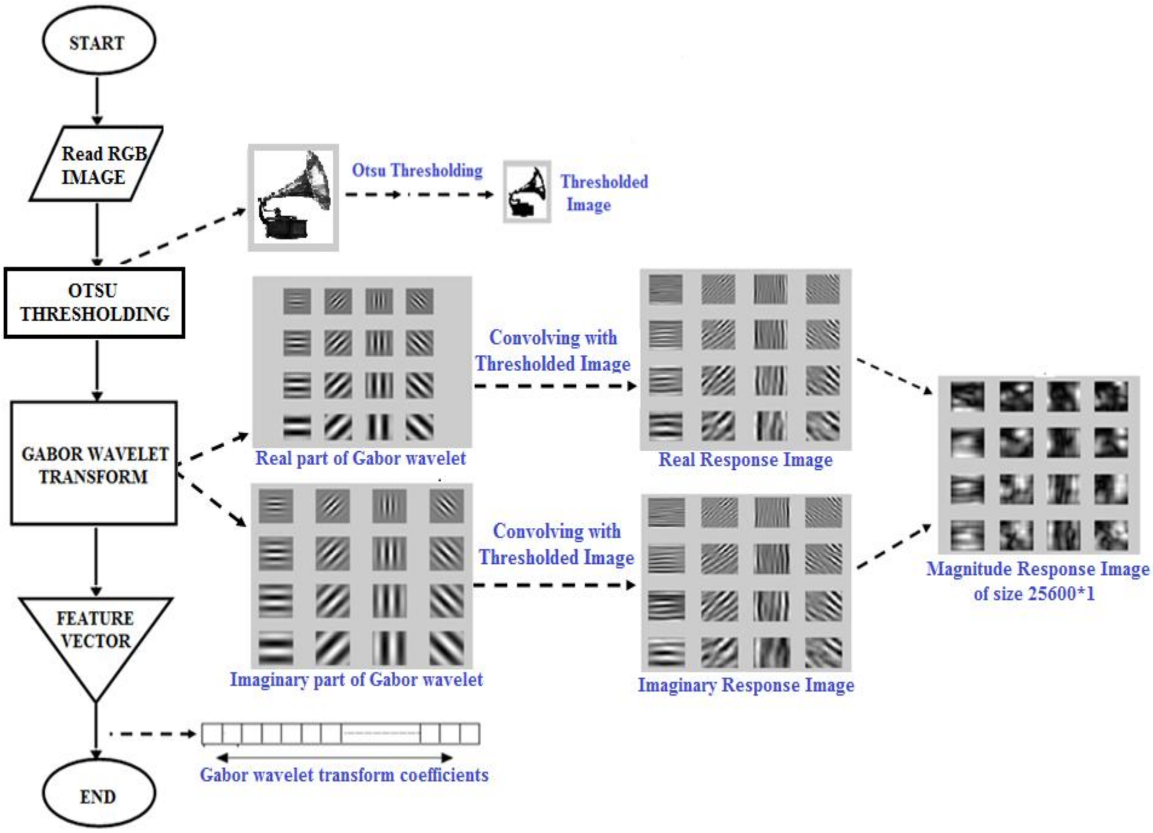

START
Read RGB IMAGE
OTSU THRESHOLDING
GABOR WAVELET TRANSFORM
FEATURE VECTOR
END
Otsu Thresholding
Thresholded Image
Real part of Gabor wavelet
Convolving with Thresholded Image
Real Response Image
Imaginary part of Gabor wavelet
Convolving with Thresholded Image
Imaginary Response Image
Magnitude Response Image of size 25600*1
Gabor wavelet transform coefficients

Figure 5.4: Flowchart for feature extraction stages of Object Images.

Step 4: As described in chapter 2 Gabor filter contains real and imaginary parts. So kernel designed is composed of real and imaginary parts with 4 orientations and 4 scales (shown in Figure 5.5 (a) and 5.5 (b)).

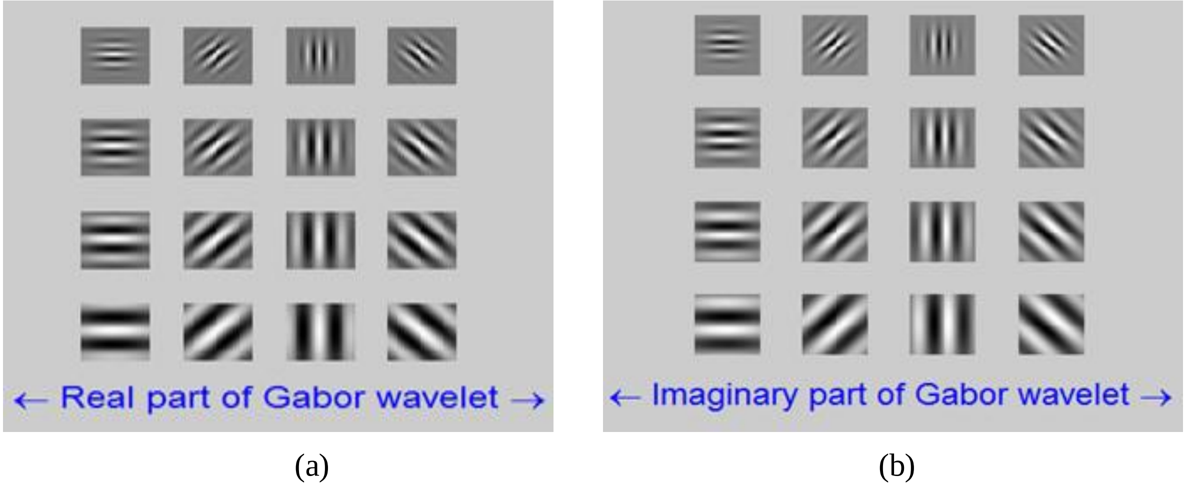
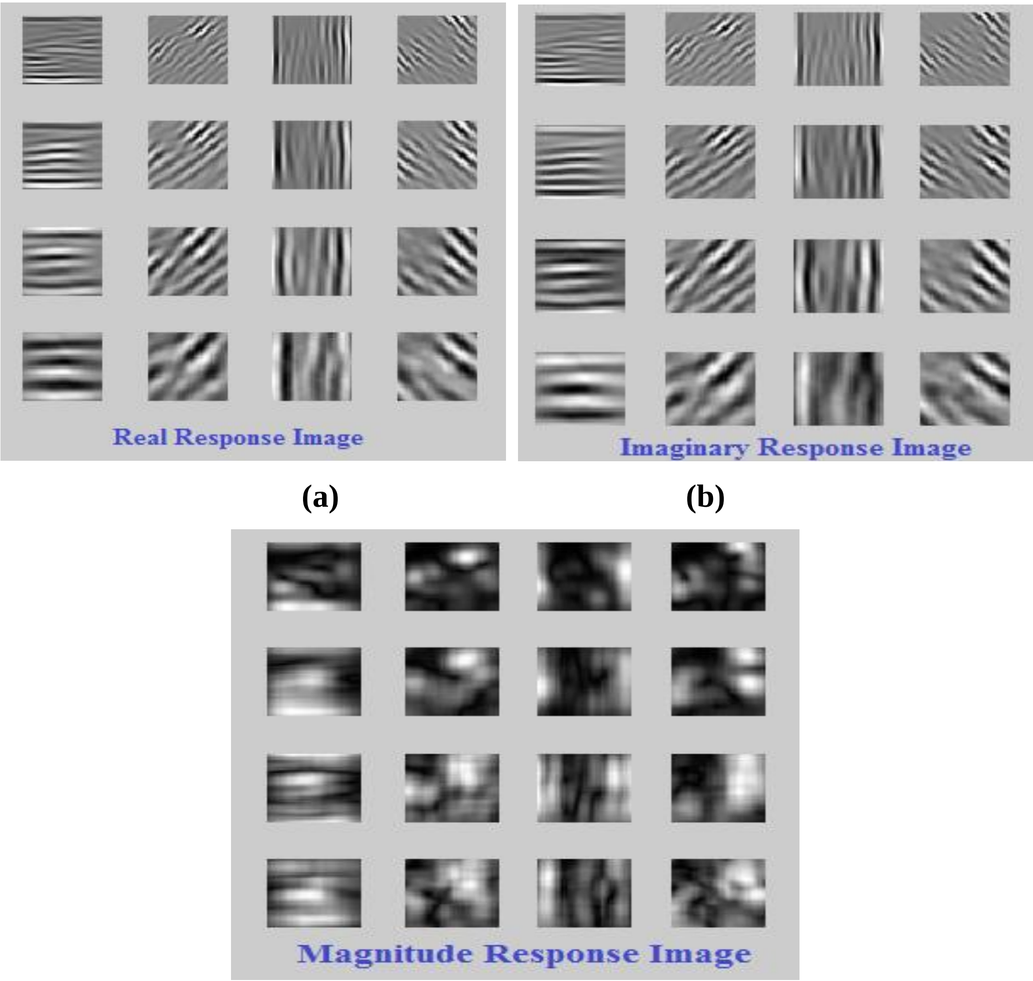

← Real part of Gabor wavelet →
← Imaginary part of Gabor wavelet →
(a)
(b)

Figure 5.5 (a): The real part of 4 × 4 Gabor wavelets. **(b):** The imaginary part of 4 × 4 Gabor wavelets.

Step 5: Then convolving the image with 16 Gabor wavelets i.e. with real and imaginary part of Gabor filter separately and obtaining 16 real and 16 imaginary responses respectively shown in Figure 5.6 (a) & (b). Here while convolution, kernel window is moving on our image with one step-size and using the zero- padding and returns the central part of the convolution of the same size as that of an image.

Step 6: After that calculating 16 magnitude responses (shown in Figure 5.6 (c)) using real and imaginary responses obtained from step 5.



(a) **(b)**

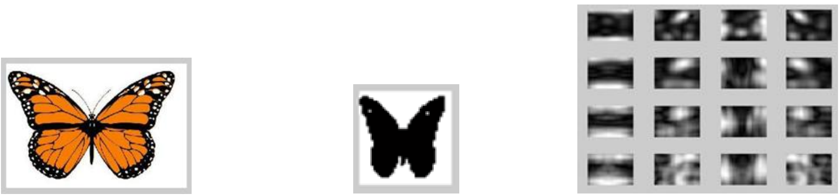
(c)

Figure 5.6 Responses of 4 × 4 Gabor wavelets. **(a):** Real Response
(b): Imaginary Response **(c):** Magnitude Response

Step 7: Repeat the above steps for all the images. So feature vector of size 25600×1 (4×4×40×40) containing magnitude response corresponding to each image is obtained.

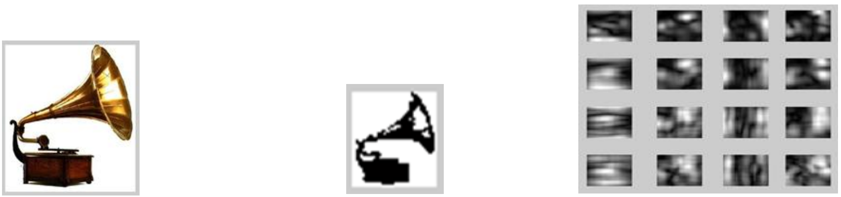
These steps are repeated for all the images used for training, as 150 images are used for training so feature matrix of size 25,600 × 150 is obtained for training. Results of feature extracted of some images are shown in Figure 5 (a)-(j) showing an image of each object's class, their corresponding thresholded images (obtained after Otsu thresholding) and Gabor features image (which is a magnitude response and is obtained after applying Gabor wavelet).

We use the RGB color images of objects for testing. So in step 1 we don’t have to convert an image to gray-scale but resizing the image to 40 × 40 × 3. Then Otsu thresholding is applied on each component separately. Then each step described above is applied on each component separately. Finally we obtain vector of size 25600×3 for each image. This way feature matrix is obtained for testing images. This feature matrix is used for recognition. After creating the feature matrix for training and testing, we applied fuzzy classifier for recognition as described in earlier chapter.

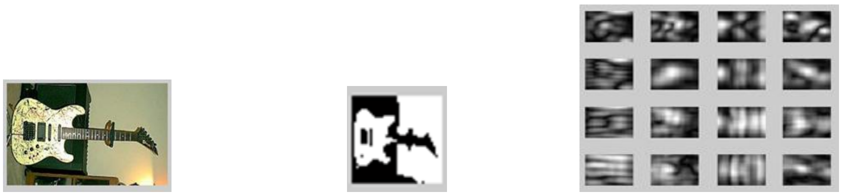


(a)

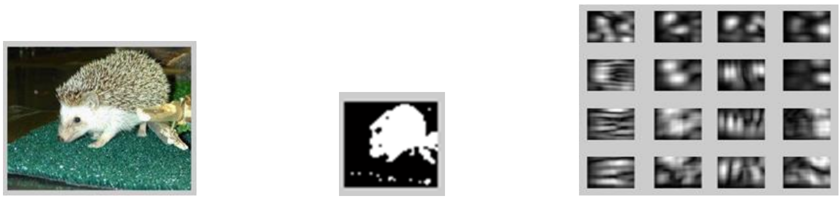
(b)



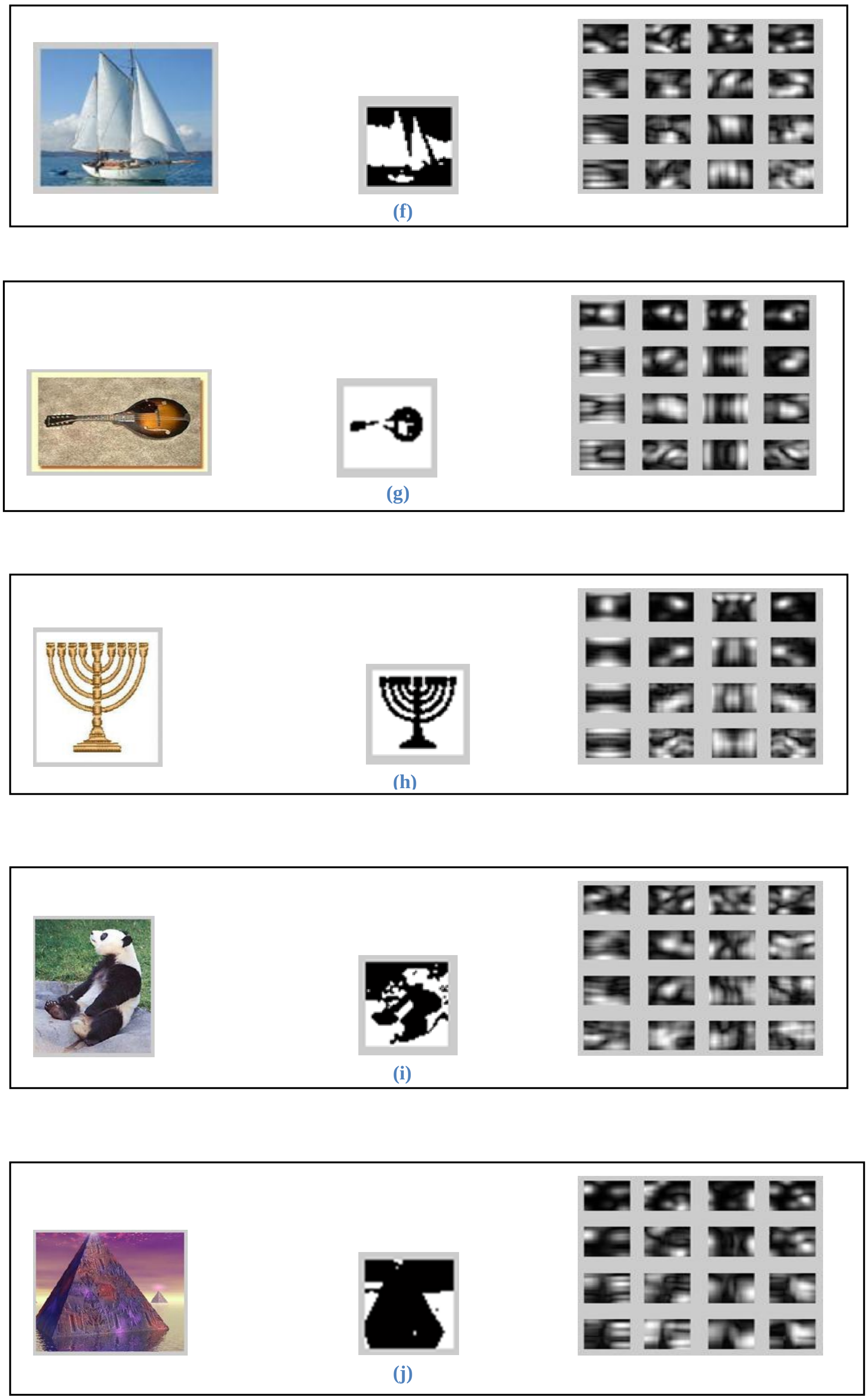
(c)



(d)



(e)



(f)

(g)

(h)

(i)

(j)

Figure 5.7: Object Classes and Their Corresponding Images i.e. Object Image, Thresholded Image (obtained from Otsu thresholding), and Magnitude Response Image of size 25600*1 (which are Gabor features obtained after applying Gabor wavelet). These results obtained are used for training. We have included objects: (a) Butterfly, (b) Ketch, (c) Garfield, (d) Gramophone, (e) Electric Guitar, (f) Hedgehog, (g) Mandolin, (h) Menorah, (i) Panda, and (j) Pyramid with their corresponding images.

CHAPTER 6

RESULTS & DISCUSSION

This chapter firstly presents the methodology used in algorithm testing, means how we set the parameter value for our proposed classifier. Then results are compared with other systems and comparison is shown for each object class. Then some of easy and difficult cases are shown and described. Finally various evaluation of results is defined which describes on what criteria evaluation for the performance of overall system can be done.

6.1 Determination of exponent of Generalized Gaussian membership-function for Fuzzy Classifier?

One possible definition of a fuzzy classifier is given in [69] as ‘Any classifier that uses fuzzy sets or fuzzy logic in the course of its training or operation’. In fuzzy logic, the membership function of a fuzzy set represents the degree of truth as an extension of valuation. In our work we have used generalized Gaussian member function as seen in Equation 4.3. We have used the value of b parameter as 1.66 in chapter 4. To show why we took b (in Equation 4.4) value as 1.66 we have trained our system with 5 images and tested our proposed classifier with 5 known images (means whose class results we already knew). We have tested all images by changing ‘b’ parameter value from 1.2 to 2 (i.e. we did coarse testing) with an interval of 0.1 (we choose 1.2 and 2 as it is the popular choice) and found the results as shown in Table 6.1.

TABLE 6.1: Classification of Training Dataset for the Proposed System

Object Classes	**b parameter value (taken with interval 0.1)**								
	1.20	**1.30**	**1.40**	**1.50**	**1.60**	**1.70**	**1.80**	**1.90**	**2.00**
1	0	0	1	4	5	5	5	5	5
2	0	0	0	1	5	5	4	4	4
3	0	0	2	4	5	5	4	4	3
4	0	0	0	2	5	4	4	3	3
5	0	0	0	0	2	3	3	3	3
6	0	0	0	3	5	5	5	5	5
7	0	0	0	2	4	5	5	5	5
8	5	5	5	4	4	3	3	3	3
9	0	0	5	5	4	3	1	1	1
10	0	0	0	3	3	4	4	4	4
Total Correctly Classified Images (out of 50)	**5**	**5**	**14**	**28**	**42**	**42**	**38**	**37**	**36**

From this table we can see how many out of 5 object sample for each object class can be correctly classified. At points i.e. ‘b’ values are 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, and 2.0 the results obtained (means total correctly classified images) are 5, 5, 14, 28, 42, 42, 38, 37, and

36 respectively. We found that higher percent of results can be obtained with b=1.6, 1.7 and 1.8 i.e. 84%, 84%, 76%. Then we analyze the results in detail, i.e we again tested same images for ‘b’ value from 1.6 to 1.8 (fine testing) with an interval of 0.01 i.e. 1.61, 1.62, 1.63, 1.64, 1.65, 1.66, 1.67, 1.68, 1.69, 1.70, 1.71, 1.72, 1.73, 1.74, 1.75, 1.76, 1.77, 1.78, and 1.79 and got the results as 41, 41, 42, 42, 43, 44, 42, 42, 42, 42, 42, 42, 42, 40, 40, 40, 40, 40, and 39 respectively, detailed view is shown in Table 6.2. We found that best results can be obtained at a value 1.66 i.e. 88% of object samples were correctly classified. So, we may conclude that taking the ‘b’ parameters value as 1.66, the proposed fuzzy classifier provides the best results at this point for any test dataset.

TABLE 6.2: Classification of Training Dataset for Proposed System with Specific Values

Object Classes	**b parameter value (taken with interval 0.01)**																		
	1.61	**1.62**	**1.63**	**1.64**	**1.65**	**1.66**	**1.67**	**1.68**	**1.69**	**1.70**	**1.71**	**1.72**	**1.73**	**1.74**	**1.75**	**1.76**	**1.77**	**1.78**	**1.79**
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4
3	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	2	2	3	3	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3
6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5
8	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
9	4	4	4	4	4	4	3	3	3	3	3	3	3	2	2	2	2	2	2
10	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Total Correctly Classified (out of 50)	**41**	**41**	**42**	**42**	**43**	**44**	**42**	**42**	**42**	**42**	**42**	**42**	**42**	**40**	**40**	**40**	**40**	**40**	**39**

6.2 Discussion on results and comparison with other classifiers

We have performed our experiments on images from Caltech dataset. We experiment our work using 32-bit OS, MATLAB 7.9.0 and processor used is Intel (R) Core (TM) 2 Duo CPU T6570 with a speed of 2.10 GHz. Here we are presenting the extensive performance evaluation of various recognition methods. In our work we have compared the results obtained from [47] using their code present at [48], in which G. Qiu presented a method to compute the statistics of achromatic and chromatic spatial patterns of colour images for indexing and content-based retrieval. For comparison we have also taken features obtained from [47] and classified with MLP-Neural Network. So, the features extracted from [47] and are fed to MLP-Neural Network classifier. The results as shown in Table 3, which shows results obtained from MLP-Neural Network are better compare to their [47] own features and classifier. We have experimented and compared with classifiers popularly used for pattern recognition problem, classifiers such as MLP-Neural Network [49,65], SVM [61-64], and Naive Bayes [57-58], Nearest Neighbour classifier [55-56] and K- Nearest Neighbour [59-60]. K- Nearest Neighbour uses Euclidean distance in our work so we treat it as Nearest Neighbour classifier.

The experimental results obtained after testing from various techniques: Color Pattern Recognition [47], Color Pattern Recognition [47] using MLP-Neural Network Classifier, Gabor Wavelet Features from Gray-scale testing images using MLP-Neural Network Classifier, Otsu & Gabor Wavelet Features from Gray-scale testing images using MLP-Neural Network Classifier, Otsu & Gabor Wavelet Features from Gray-scale testing images using Nearest Neighbour Classifier, Otsu & Gabor Wavelet Features from Gray-scale testing images using SVM Classifier, Otsu & Gabor Wavelet Features from Gray-scale testing images using Naive Bayesian Linear Classifier, Otsu & Gabor Wavelet Features from Gray-scale testing images using proposed Fuzzy Classifier, Otsu & Gabor Wavelet Features from RGB testing images using Nearest Neighbour Classifier, Otsu & Gabor Wavelet on RGB and with RGB Training set using Nearest Neighbour Classifier and Otsu & Gabor Wavelet Features from RGB testing images using proposed Fuzzy Classifier for ten object classes such as Butterfly, Ketch, Garfield, Gramophone, Electric Guitar, Hedgehog, Mandolin, Menorah, Panda, and Pyramid were evaluated. The computation time was also noted. The operational details about all the classifiers have been discussed in the above chapters. The obtained results have been depicted in Table 6.3. Results of our proposed classifier can be seen in last column of Table 6.3. These results have been compared with various other systems results and shows that our system performs better than all other systems. To evaluate the performance we calculated true positive which is when an image containing the object is identified as containing the object class. Therefore efficiency can be calculated from true positive rate which is also called as Recognition rate can be defined as:

$$\text{Recognition Rate} = \frac{\text{Number of true positives}}{\text{Number of images containing the object class}} * 100 \qquad \textbf{..... (6.1)}$$

To further evaluate the performance of our new method, we have analyzed the results obtained in more detail i.e. for each object class (we presented the recognition rate for each class in decreasing order).

- The measurement for conclusion of **Pyramid** object class provides the excellent performance of our system in comparison of others i.e. its efficiency is **90.5%** followed by Otsu & GW on RGB and with RGB Training set using Nearest Neighbour Classifier with performance of 78.6%, Otsu & GW on RGB with Nearest Neighbour Classifier, and Otsu & GW on Gray with MLP-Neural Network Classifier which gives performance of 73.8% each, GW only on Gray with MLP-Neural Network Classifier, Otsu & GW on Gray with Nearest Neighbour Classifier, Otsu & GW on Gray with Proposed Fuzzy Classifier, and Otsu & GW on Gray with Naive Bayesian Linear Classifier which gives performance of 66.7%, 64.3%, 61.9%, and 57.1% respectively and rest all other systems performs less than 48%.

- Let's we juxtapose the resultant of **Ketch** object class then we observe that an efficiency of our method goes to **82.8%** which is in virtuous position among other methods. Other systems Otsu & GW on Gray with MLP-Neural Network Classifier, GW only on Gray with MLP-Neural Network Classifier, Otsu & GW on RGB and with RGB Training set using Nearest Neighbour Classifier, Otsu & GW on Gray with Proposed Fuzzy Classifier, Otsu & GW on Gray with Nearest Neighbour Classifier, and Otsu & GW on RGB with Nearest Neighbour Classifier gives efficiency of 75.8%, 73.7%, 71.7%, 67.7%, 66.7% and 65.7% respectively and rest all other systems gives performance less than 60%.
- When we analyzed the results of **Mandolin** object class we found that our system performs marvellously well with an efficiency of **78.6%** better than all other systems while Otsu & GW on Gray with MLP-Neural Network Classifier, Otsu & GW on Gray with Nearest Neighbour Classifier, Otsu & GW on Gray with SVM Classifier, Otsu & GW on Gray with Proposed Fuzzy Classifier gives performance of 75%, 64.3%, 60.7%, 60.7% respectively and other systems less than 58%.
- We relate the conclusion of **Butterfly** object class then we find that our system's procedure provide higher efficiency of **75%** in comparison of other systems, in-fact systems such as Otsu & GW on RGB with Nearest Neighbour Classifier, Otsu & GW on Gray with Proposed Fuzzy Classifier, Otsu & GW on Gray with MLP-Neural Network Classifier, Otsu & GW on Gray with Nearest Neighbour Classifier, Otsu & GW on RGB and with RGB Training set using Nearest Neighbour Classifier and GW only on Gray with MLP-Neural Network Classifier gives an efficiency of 65.8%, 64.5%, 60.5%, 59.2%, 55.3% and 53.9% respectively and rest all other system gives efficiency of less than 39%.
- When we compare the results of **Menorah** object class we can see that our system gives an efficiency of **73.6%** and an equivalent efficiency is given by system GW only on Gray

with MLP-Neural Network Classifier while Otsu & GW on Gray with Nearest Neighbour Classifier, Otsu & GW on Gray with Proposed Fuzzy Classifier and Otsu & GW on Gray with MLP-Neural Network Classifier gives 72.2%, 70.8%, and 66.7% and rest others gives less than 58%.

- If the result is distinguishable of **Electric-Guitar** object class then it can be observed that it improves the efficiency of system and that is much better than other techniques, which is **71.7%** followed by Otsu & GW on Gray with Proposed Fuzzy Classifier, Otsu & GW on Gray with Nearest Neighbour Classifier, Otsu & GW on RGB and with RGB Training set using Nearest Neighbour Classifier and Otsu & GW on RGB with Nearest Neighbour Classifier which gives an efficiency of 65%, 55%, 51.7% and 50% respectively but other systems gives performance less than 47%.
- If we differentiate the result of **Gramophone** object class then we notice that our technique gives an efficiency of **63.9%** while other systems like GW only on Gray with MLP-Neural Network Classifier, and Otsu & GW on Gray with Proposed Fuzzy Classifier shows 58.3% and 55.6% efficiency respectively and rest all other systems shows performance less than 39%, which is much greater than other techniques.
- When the results of **Garfield** object class were differentiated then it can be observed that our systems performance is **63.2%** and an equivalent performance can be analyzed with Otsu & GW on Gray with Proposed Fuzzy Classifier system while Otsu & GW on RGB with Nearest Neighbour Classifier system gives an efficiency of 52.6% and rest all other systems gives a performance less than 48%.
- If the results of **Hedgehog** object class were analyzed it was found that our systems performance is **61.5%** while the systems Color Pattern Recognition [47] with MLP-Neural Network Classifier and Otsu & GW on Gray with Naive Bayesian Linear Classifier gives efficiency **66.7%** and 64.1% respectively better than our system while rest other systems gives efficiency less than our system.
- We compare the results of **Panda** object class then we analyzed that our system efficiency is **52.2%** and equivalent performance is given by Otsu & GW on Gray with Proposed Fuzzy Classifier system but GW only on Gray with MLP-Neural Network Classifier system gives performance of **56.5%** which is better than our system while all other systems gives a performance less than our system and less than 48%.

From the above discussion, it can be analyzed that our proposed system is 15.14%, 14.93%, & 14% more efficient than the best performance achieved from any other compared system for the class Pyramid, Garfield & Butterfly respectively. Whereas for the object class Panda & Hedgehog it is observed that our system is just 7.6% & 7.8% less efficient than the best performance achieved from any other compared system. Hence it can be concluded that our fuzzy classifier has shown magnificent results for most of the object classes. Thus it would be worth noting that our system outperforms as compared to all other systems. So, without doubt, the proposed fuzzy classifier is the best technique out of all, giving the overall Recognition Rate of is 74.1%.

6.2.1 Pros and Cons of different Classifiers: ***Nearest neighbour classifiers*** are based on minimum distance metric. **Pros:**

- Simple to implement.
- Flexible to feature / distance choices.
- Easy to compute.
- Straightforward logic.
- Naturally handles multi-class cases.
- Can do well in practice with enough representative data.
- Fast training.
- Excellent performance on a wide range of tasks.

Cons:

- Large search problem to find nearest neighbours.
- Basic method.
- Storage of data.
- Must know we have a meaningful distance function.
- It is not sensitive to complex patterns unless extended to include more than just nearest neighbours.

SVM classifiers were relatively new concept having nice generalization properties. They are hard to learn – learned in batch mode using quadratic programming techniques. Using kernels can learn very complex functions. **Pros:**

- Many publicly available SVM packages: http://www.kernel-machines.org/software.
- Kernel-based framework is very powerful, flexible.
- SVMs work very well in practice, even with very small training sample sizes.
- Often a sparse set of support vectors – compact at test time.

Cons:

- No “direct” multi-class SVM, must combine two-class SVMs.
- Computation, memory.
- During training time, must compute matrix of kernel values for every example pair.
- Learning can take a very long time for large-scale problems.
- Can be tricky to select best kernel function for a problem.

Naïve Bayes Classifiers are based on Bayesian classification. **Pros:**

- Easy to implement.
- Good results obtained in most of the cases.
- Fast.
- Induced classifiers are easy to interpret.
- Robust to irrelevant attributes.
- Uses evidence from many attributes.

Cons:

- Assumption: class conditional independence, therefore loss of accuracy.
- Practically, dependencies exist among variables. E.g., hospitals: patients: Profile: age, family history etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc.
- Dependencies among these cannot be modeled by Naïve Bayesian Classifier.
- Low performance ceiling on large databases.

Neural Network Classifier: It is a quiet old technique. It generalizes well but doesn’t have strong mathematical foundation. It can easily be learned in incremental fashion. To learn complex functions – use multilayer perceptron (not that trivial). **Pros:**

- Can learn more complicated class boundaries.
- Fast application.
- Can handle large number of features.

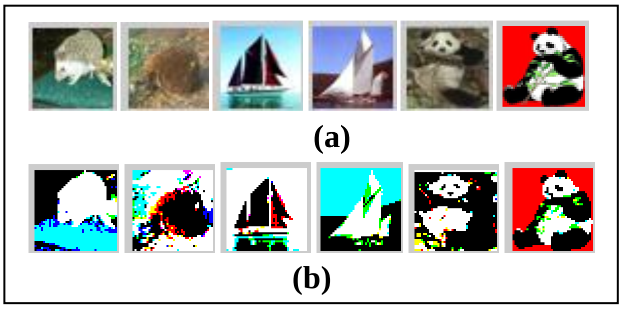
Cons:

- Slow training time.
- Require Rigorous training.
- Different results are produced after every time we train.
- Hard to interpret.
- Hard to implement: trial and error for choosing number of nodes.

6.3 Some Easy and Difficult Cases

There are cases of objects which are easy to recognize as well there are cases which were difficult to recognize or were unable to recognize. As discussed above you can see that Panda class and Hedgehog class were least classified but Ketch was highly classified. So we have taken some examples of these three classes for description of easy object image cases and difficult object image cases but were correctly classified and wrongly classified cases.

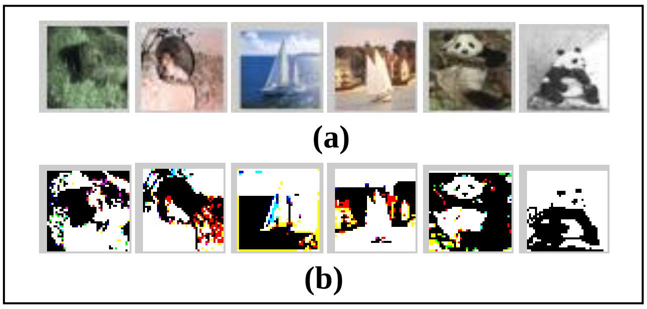
Easy Cases: Some examples of easy cases are shown in Figure 6.1 which shows that these images were easy to recognize because in these cases objects are very well distinguished from its background. The features of the object collected are able to completely describe the object. From Figure 6.1 (b) we can see that in Otsu Images Hedgehog, Ketch and Panda can be easily recognized.


(a)

(b)

Figure 6.1 (a): Object Images of Easy Cases showing from left to right 2 images each of Hedgehog, Ketch, and Panda Object Classes. **(b):** Their corresponding Otsu Images in 3 D.

Difficult Cases but were Correctly Classified: The difficult cases for object recognition are shown in Figure 6.2. The objects shown in Figure 6.2 (a) are distinguished but with great difficulty. The image of objects were distinct but were difficult to recognize. From Otsu Images of 6.2 (a) as shown in 6.2 (b) we can see that all these images are difficult to recognize.


(a)

(b)

Figure 6.2 (a): Object Images of Difficult Cases which were correctly classified showing from left to right 2 images each of Hedgehog, Ketch, and Panda Object Classes.
(b): Corresponding Otsu Images of 6.2 (a) in 3 D.

First two images are of Hedgehog object class from their corresponding Otsu images it is difficult to recognize the objects by many of the classifiers but were successfully recognized by our proposed classifier. Next Image is of Ketch object class but from its Otsu image it seems to be of Pyramid object class and was not correctly classified by some of classifiers. Next image is again of Ketch object class, it does not contain high features in its Otsu image that's the reason it was not correctly classified. Next two images are of Panda object class. From their corresponding Otsu image 1st image of panda seems to be of Garfield object class where as next image contains very low features for not being correctly recognized. Finally we can say that it is due to Otsu thresholding we are able to efficiently recognize the objects. So it is an essential step which tries to extract the shape of an object from an image.

Wrongly Classified Cases: Where as the wrongly classified cases for object recognition are shown in Figure 6.3, which shows that these objects are not distinguished. There is merging of objects characteristics with the background either due to background having very high features or due to objects having very low features which makes recognition of object difficult. In Figure 6.3 (a) we have shown object images of Hedgehog, Ketch and Panda which were not recognize and in 6.3 (b) their corresponding Otsu images are shown, 6.3 (c) shows the corresponding images of those objects from training dataset to which the degree of membership is high, and 6.3 (d) shows the Otsu image of those classified object images.

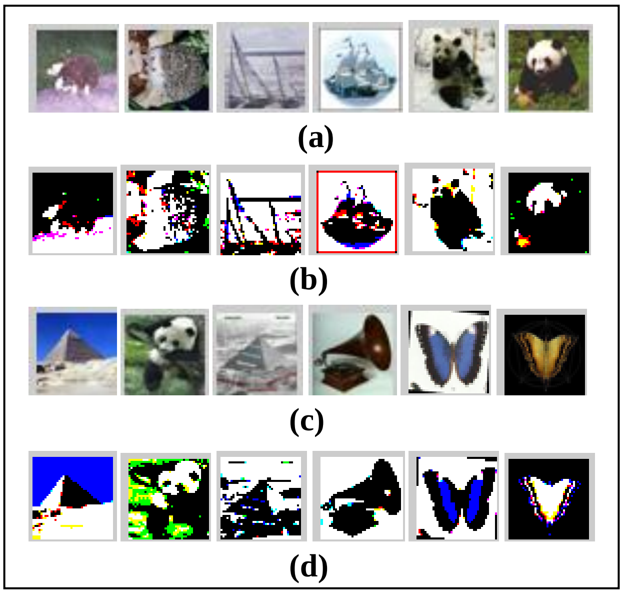


Figure 6.3 (a): Object Images for Wrongly Classified Cases showing from left to right 2 images each of Hedgehog, Ketch, and Panda Object Classes. **(b):** Corresponding Otsu Images of 6.3 (a) in 3-dimension. **(c):** The Classes to which 6.3 (a) got classified. **(d):** Corresponding Otsu Images of 6.3 (c) in 3-dimension.

As the first image of Hedgehog shown as 1st image in Figure 6.3 (a) was not correctly classified and it was classified to Pyramid object class, it is visible from both Otsu image of object to recognized and Otsu image of classified object image that there are high features in upper portion but very low features in lower portion. Next image of Hedgehog shown as 2nd image in Figure 6.3(a) also was not correctly recognized from its Otsu image you will be able to observe the object is not visible, this object have high degree of membership for Panda object class image. Similarly first image of Ketch shown as 3rd image in Figure 6.3 (a) was not correctly classified and its Otsu image shows that even the object is also not clearly visible, this object is classified to Pyramid object class image. Next image of Ketch shown as 4th image in Figure 6.3 (a) has high degree of membership for Gramophone object class image and from Otsu image of object we are not able to make decision what object is it, and its observed from both of its Otsu images that there are object features are present in center of the images. Both the images of Panda shown as 5th and 6th image in Figure 6.3 (a) were classified to Butterfly object images. In both objects Otsu image shows that object is also not clearly distinguished, and from Otsu images of first panda shows that dark portion is at center where as second image shows that light portion is at center.

6.4 Evaluation of Results

The evaluation criteria are used to evaluate the performance of the object recognition methods and to build a common basis that facilitates an objective comparison. The description and the analysis of various evaluation criteria, i.e., first the results and discussion of the robustness, then the accuracy, and finally the computation time are presented.

The first criterion to be considered is the robustness of the approach. This includes the robustness against occlusions, e.g., caused by overlapping objects on the assembly line. Non-linear as well as local illumination changes are also crucial situations, which cannot be avoided in many applications over the entire field of view. We measure the robustness using the recognition rate, which is defined as the number of images in which the object was correctly recognized divided by the total number of images.

We perform the same experiments on various systems such as Color Pattern Recognition [47], Color Pattern Recognition [47] using MLP-Neural Network Classifier, Gabor Wavelet Features from Gray-scale testing images using MLP-Neural Network Classifier, Otsu & Gabor Wavelet Features from Gray-scale testing images using MLP-Neural Network Classifier, Gabor Wavelet Features from Gray-scale testing images using Nearest Neighbour

Classifier, Gabor Wavelet Features from Gray-scale testing images using SVM Classifier, Gabor Wavelet Features from Gray-scale testing images using Naive Bayesian Linear Classifier, Gabor Wavelet Features from Gray-scale testing images using proposed Fuzzy Classifier, Gabor Wavelet Features from RGB testing images using Nearest Neighbour Classifier, and Gabor Wavelet Features from RGB testing images using proposed Fuzzy Classifier i.e. we used same dataset for all the systems for object recognition task and found that 23.3%, 38.1%, 58.7%, 60.9%, 57.9%, 46.4%, 45%, 63.4%, 56.9%, and 74.1% samples respectively were correctly classified. Recognition rates are shown in Table 6.3

The second criterion is the accuracy of the methods. Most applications need the exact transformation parameters of the object as input for further investigations like precise metric measurements. Our system uses Gabor wavelet, which have good characteristics of orientation and space-frequency localization i.e. their main aim is to use their multi-resolution, multi-orientation properties and are commonly used for extracting local features for various applications like object detection, recognition and tracking. So it ensures a reliable recognition of objects at various orientations as well as at various scales.

The computation time represents **the third evaluation criterion**. Despite the increasing computation power of modern microprocessors, efficient and fast algorithms are more important than ever. This is particularly true in the field of object recognition, where a multitude of applications enforce real time computation. Indeed, it is very hard to compare different recognition methods using this criterion because the computation time strongly depends on the individual implementation of the recognition methods. Nevertheless, we tried to find time taken by each of the investigated approaches in-order to at least allow a qualitative comparison. Our system just takes approx. 6.5 seconds for each sample object image to get classify. To calculate the computation time for all the classifiers except MLP-Neural Network is done by a coding command while for MLP-Neural Network as we used the GUI tool so we did it through stop watch. Computation time of all systems can be seen in Table 6.3.

TABLE 6.3: Comparison of Proposed System with Other Systems for 494 test images

SYSTEM **OBJECT CLASS**	**Total Images per class (for tests)**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**A11**
		(%age Correctly Classified)										
1. Butterfly	**76**	30.3	36.8	53.9	60.5	59.2	38.2	31.6	64.5	65.8	55.3	**75**
2. Garfield	**19**	42.1	42.1	0	31.6	47.4	26.3	26.3	63.2	52.6	42.1	**63.2**
3. Gramophone	**36**	30.6	16.7	58.3	33.3	36.1	19.4	33.3	55.6	38.9	38.9	**63.9**
4. Electric Guitar	**60**	18.3	23.3	46.7	46.7	55.0	33.3	38.3	65	50	51.7	**71.7**
5. Hedgehog	**39**	43.6	**66.7**	59.0	59.0	48.7	56.4	64.1	51.3	51.3	56.4	61.5
6. Ketch	**99**	19.2	56.6	73.7	75.8	66.7	59.6	51.5	67.7	65.7	71.7	**82.8**
7. Mandolin	**28**	14.3	17.9	35.7	75.0	64.3	60.7	46.4	60.7	57.1	50	**78.6**
8. Menorah	**72**	15.3	34.7	73.6	66.7	72.2	59.7	48.6	70.8	57	55.6	**73.6**
9. Panda	**23**	26.1	17.4	**56.5**	47.8	17.4	30.4	43.5	52.2	17.4	26.1	52.2
10. Pyramid	**42**	11.9	38.1	66.7	73.8	64.3	47.6	57.1	61.9	73.8	78.6	**90.5**
Average (%age)		**23.3**	**38.1**	**58.7**	**60.9**	**57.9**	**46.4**	**45.0**	**63.4**	**56.9**	**56.9**	**74.1**
Time per sample (in seconds)		**0.22**	**28.73**	**76.24**	**77.59**	**0.75**	**6.19**	**4.37**	**2.09**	**0.56**	**0.09**	**6.54**

where: A1 is Color Pattern Recog. [47] by using indexing coloured image patterns (RGB training, RGB testing)

A2 is Color Pattern Recog.[47], MLP-Neural Network Classifier (RGB training, RGB testing)

A3 is GW only on Gray, MLP-Neural Network Classifier (Gray training, Gray testing)

A4 is Otsu & GW on Gray, MLP-Neural Network Classifier (Gray training, Gray testing)

A5 is Otsu & GW on Gray, Nearest Neighbour Classifier (Gray training, Gray testing)

A6 is Otsu & GW on Gray, SVM Classifier (Gray training, Gray testing)

A7 is Otsu & GW on Gray, Naive Bayesian Linear Classifier (Gray training, Gray testing)

A8 is Otsu & GW on Gray, Proposed Fuzzy Classifier (Gray training, Gray testing)

A9 is Otsu & GW on RGB, Nearest Neighbour Classifier (Gray training, RGB testing)

A10 is Otsu & GW on RGB and with RGB Training set, Nearest Neighbour Classifier (RGB training, RGB testing)

A11 is Otsu & GW on RGB, Proposed Fuzzy Classifier

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Object recognition is the subfield of computer vision whose main task is to identify or locate a set of objects from an image or video sequence. Object Recognition is one of the most important, yet least understood aspects of visual perception. It is not restricted to single class of objects for example face recognition, fingerprint recognition or vehicle recognition. Rather, it involves object recognition from novel views of multiple categories of objects for which some training samples are available. Recognition domains where the exact features distinguishing one class of objects from others are unknown have revived our interest in this field of object recognition. As mentioned above Recognition can be done by image-based and feature-based methods and we chose Image based approach in which object models can be compared directly or fairly directly with input data. A great advantage of this method is that any shape can be represented no matter how complex as long as we can take images of it. For feature extraction we first segment our image with thresholding technique using Otsu method then we used texture based approach and model based method i.e. Gabor wavelet extraction method. In the past it had been used for object detection, object recognition, object tracking, face tracking, face recognition, optical character recognition, iris recognition, fingerprint recognition, and texture analysis. Gabor wavelets exhibit desirable characteristics of spatial locality and orientation selectivity. As described earlier this approach has several advantages against robustness, illumination, multi-resolution, and multi-orientation. These extracted features are used for classification. In this thesis a new classifier is proposed which is capable of recognizing the object features very efficiently. For training and testing Caltech dataset was used. Comparisons were made with many other techniques as explained earlier and results shown in Table 6.3. The reasons our approach is important for model-based object recognition as well other conclusions can be made are:

- Our system does not use any image segmentation to extract objects from the background.
- It does not need explicit shape features and contours which cannot be reliably detected.
- It can tolerate significant amount of object distortions due to viewing geometry, scale, aspect and environmentally induced deviations.
- It allows us to recognize objects at different scales since we can estimate the scale of an object using the multi-scale Gabor wavelet representation of an object model.
- The overall robustness i.e. Recognition rate achieved by the proposed fuzzy classifier is higher than all the other techniques that are commonly used for object recognition.

- The computation time taken by the proposed fuzzy classifier is very less i.e. it takes approximately 6 seconds to test single sample
- The proposed fuzzy classifier performs very well even when we used it for Gray test images as you can see results of A8 in Table 6.3.
- The low complexity of the design and the low cost of implementation of the system make this technique a very feasible one for practical purposes.

The new proposed system for object recognition which is based on recognition of Otsu with Gabor features thresholding of RGB using proposed fuzzy classifier works well. The results were shown for each object which shows that the Performance of our system outperforms in maximum of the classes compared to other systems. So it's worth noting that our proposed fuzzy classifier gives the best results compared to other techniques. The overall Recognition Rate is 74.1%. So the proposed technique can be used for real-time applications.

7.2 Future Work

Object Recognition plays an important role in today's world. As defined in chapter 1 object recognition can be used for various applications such as Robotics, Face recognition and so on which means that this technique can be used for security purposes which require the Recognition Rate must be as high as possible rate and minimum misclassification rate. In this regard, the results obtained by the proposed fuzzy classifier were found to be excellent and much better as compared to all the other techniques as explained earlier. But as future work we will try to extend this work in order to reduce the misclassification rate. So, one of the biggest challenges for the future remains to improve the performance of the system as to achieve exceptionally good results in order to make our system much reliable. In this regard, as observed in chapter 6 we would try to improve the results by using different binarizing method which can segment the object in such a way that we will be able to very well distinguish the object from its background. As in our work we had determined the exponent of Generalised Gaussian membership function experimentally but in future we would like to give a mathematical approach for that either by using graphical method or by any other methods. Another goal is to increase the speed of our system, as our system makes the feature vector too bulky means for an image of size 40 x 40 the feature vector size is 25600 x 1 which is greater compared to some other system. So if we could be able to reduce the size of feature vector, definitely the speed of our proposed classifier will get increase making the real time applications to classify or we can say to take decisions much faster.

REFERENCES:

[1] Roach, J. W., Aggarwal, J. K., "Computer Tracking of Objects Moving in Space", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-1, issue 2, pp. 127 – 135, 1979.

[2] Yoram Yakimovsky, "Boundary and object detection in real world images", *IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes,* vol.: 13, part: 1, 1974.

[3] N. E. Nahi, S. Lopez-Mora, "Estimation of object boundaries in noisy images", *IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes,* vol.: 15, part: 1, 1976.

[4] M. Hanmandlu, K.R. Murali Mohan, Sourav Chakraborty, Sumeer Goyal, D. Roy Choudhury, "Unconstrained handwritten character recognition based on fuzzy logic", *Pattern Recognition,* pp. 603–623, 2003.

[5] Rolf P. Würtz, "Object Recognition Robust under Translations, Deformations, and Changes in Background", *IEEE Trans on Pattern Analysis and Machine Intelligence,* pp. 769–799, July 1997.

[6] Hasan Ugur Ozer and Dr. Ramakrishnan Sundaram, "Object Recognition with Generic Self Organizing Feature Extractors and Fast Gabor Wavelet Transform", *Proceedings of International Joint Conference on Neural Networks, Florida, USA,* August 12-17, 2007.

[7] Xing Wu and Bir Bhanu, Fellow, IEEE, "Gabor Wavelet Representation for 3-D Object Recognition", *IEEE Transactions On Image Processing,* vol. 6, no. 1, January 1997.

[8] D. Gabor, "Theory of communication", *Journal of Institute of Electrical Engineers,* vol. 93, pp.429-457, 1946.

[9] N.Otsu, "A threshold selection method from gray level histograms", *IEEE SMC-9,* pp. 62-66, Jan 1979.

[10] Caltech, http://en.wikipedia.org/wiki/Caltech_101.

[11] R. Fergus and P. Perona, "Caltech Object Category datasets", http://www.vision.caltech.edu/ Image_Datasets/ Caltech101/Caltech101.html, 2003.

[12] Linlin Shen and Li Bai, "A review on Gabor wavelets for face recognition", *Pattern Analysis Application,* vol. 9, no. 2-3, pp. 273-292, August 2006.

[13] Gabor Transform, http://en.wikipedia.org/wiki/Gabor_transform.

[14] Burcu kepenekci, "Face Recognition using Gabor Wavelets Transform", *Thesis at citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.pdf,* September 2001.

[15] Mian Zhou, "Gabor-Boosting Face Recognition", *Thesis at det.tjfsu.edu.cn/faculty /Thesis.pdf,* September 2008.

[16] Hossein Sahoolizadeh, Davood Sarikhanimoghadam, and Hamid Dehghani, "Face Detection using Gabor Wavelets and Neural Networks", *World Academy of Science, Engineering and Technology 45,* 2008.

[17] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. Von der Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the Dynamic Link Architecture," *IEEE Transactions on Computers*, vol. 42, pp. 300-311, 1993.

[18] F. Dornaika, F. Chakik, "Efficient Object Detection and Matching Using Feature Classification ", *IEEE Conference Pattern Recognition (ICPR), 2010 20th International Conference*, pp. 3073 – 3076, 2010.

[19] Li He, Hui Wang, Hong Zhang, "Object detection by parts using appearance, structural and shape features"; *IEEE Conference Mechatronics and Automation(ICMA), 2011 International Conference*; pp. 489 – 494, 2011.

[20] Zhang Xiaoyan, Liu Lingxia, Zhuang Xuchun, "An automatic video object segmentation scheme ", *IEEE Conference Intelligent Signal Processing and Communication Systems, 2007 (ISPACS), 2007 International Symposium*, pp. 272 – 275, 2007.

[21] Mei Han, A. Sethi, Wei Hua, Yihong Gong, "A detection-based multiple object tracking method", *IEEE Conference Image Processing, 2004 (ICIP '04)*, vol. 5, pp. 3065 – 3068, 2004.

[22] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.

[23] Ping-Han Lee, Yen-Liang Lin, Shen-Chi Chen, Chia-Hsiang Wu, Cheng-Chih Tsai, Yi-Ping Hung, "Viewpoint-Independent Object Detection Based on Two-Dimensional Contours and Three-Dimensional Sizes", *Intelligent Transportation Systems, IEEE Transactions*, vol. 12, issue 4, pp. 1599 – 1608, 2011.

[24] R.N. Strickland, He Il Hahn, "Wavelet transform methods for object detection and recovery", *Image Processing, IEEE Transactions*, vol. 6, issue 5, pp. 724 – 735, 1997.

[25] T. Modegi, "Small object recognition techniques based on structured template matching for high-resolution satellite images", *IEEE conference on SICE*, pp. 2168 – 2173, 2008.

[26] T. Warsop, S. Singh, "A survey of object recognition methods for automatic asset detection in high-definition video", *IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS) 2010*, pp. 1 – 6, 2010.

[27] A. Laika, W. Stechele, "A review of different object recognition methods for the application in driver assistance systems", *IEEE conference on Image Analysis for Multimedia Interactive Services 2007*, pp. 10, 2007.

[28] M. Ehrennmann, D. Ambela, P. Steinaus, R. Dillmann, "A comparison of four fast vision based object recognition methods for programming by demonstration applications", *IEEE International Conference on Robotics and Automation, 2000. Proceedings ICRA'00*, vol. 2, pp. 1862 – 1867, 2000.

[29] Vinay Kumar. B, Shreyas. B. S, “Face Recognition Using Gabor Wavelets”, *2006 Fortieth Asilomar Conference on Signals Systems and Computers, Available from ieeexplore.ieee.org*, 2006.

[30] Gareth Loy, “Fast Computation of the Gabor Wavelet Transform”, *Digital Image Computing Techniques and Application*s, *Melbourne, Australia*, 21–22 January 2002.

[31] S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, and D. Banin, “Tracking facial feature points with gabor wavelets and shape models”, *In Proc 1st International Conference on Audio- and Video-based Biometric Person Authentication,* 1997.

[32] S. Arseneau and J.R. Cooperstock, “Real-Time Image Segmentation for Action Recognition”, *Proc. IEEE Pacfic Rim Conference on Communications, Computers and Signal Processing, Victoria, B. C., Canada,* pp. 86-89, 1999.

[33] H. Tian, T. Srikanthan and K.V. Asari, “Automatic Segmentation Algorithm for the Extraction of Lumen Region and Boundary from Endoscopic Images”, *Med. Biol. Eng. Comput.,* 39(1), pp. 8-14, 2001.

[34] R. Harrabi, E. Ben Bra jek, “Color image Segmentation using automatic thresholding techniques”, *IEEE 8th International Multi-Conference on Systems, Signals & Devices,* 2011.

[35] R.M. Rangayyan, R.J. Ferrar, J.E.L. Desautels, A.F. Frere, “Directional Analysis of Images with Gabor Wavelets”, *IEEE Computer Graphics and Image Processing, Proceedings XIII Brazilian Symposium,* pp. 170 - 177, 2000.

[36] Zhongliang Fu, Chunya Tong, Huang Yan, Zhou Fan, “Parallel Gabor Wavelet Transform for Edge Detection”, *IEEE International Conference on Internet Technology and Applications,* pp. 1 – 3, 2010.

[37] Z. Yavuz, C. Kose, "Comparing 2D matched filter response and Gabor filter methods for vessel segmentation in retinal images", *IEEE 2010 National Conference on Electrical, Electronics and Computer Engineering (ELECO),* pp. 648 - 652; 2010.

[38] R. Neil Braithwaite and Bir Bhanu,“Hierarchical Gabor Filters for Object Detection in Infrared Images”, *Proceedings CVPR '94, 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1994.

[39] Like Zhang, Dong Qin,“The Object Tracking Study Based on the Gabor Wavelet Character”, *IEEE International Conference on Mechatronics and Automation,* 2006.

[40] Xing Wu and Bir Bhanu, “Gabor Wavelets for 3-D Object Recognition”, *Proceedings Fifth International Conference on Computer Vision,* 1995.

[41] M. Cheriet, J. N. Said, and C. Y. Suen, “A Recursive Thresholding Technique for Image Segmentation”, *IEEE Transactions on image processing,* vol. 7, no. 6, June 1998.

[42] Meei-Ling Hung, Chin-Shu Chang, Ji-ding Chen, Jui-Sheng Lin, Teh-Lu Liao, “Design and implementation of Real-Time Object Tracking System Using the Gaussian Motion Model and the Otsu Algorithm”, *KAM '09. Second International Symposium on Knowledge Acquisition and Modeling,* 2009.

[43] Martin Garcia Keller, “Matching Algorithms and Feature Match Quality Measures for Model-Based Object Recognition with Applications to Automatic Target Recognition”, Thesis at www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA440328, 1999.

[44] Xiaohua Zhang, Hongfeng Li, Licheng Jiao,“A change detection algorithm based on object feature for SAR image”, *IEEE APSAR 09, 2nd Asian-Pacific Conference on Synthetic Aperture Radar,* 2009.

[45] Ping Gao, Xiangju Sun, Wei Wang, “Moving Object Detection Based on Kirsch Operator Combined with Optical Flow”, *IEEE Image Analysis and Signal Processing (IASP),* 2010.

[46] Hua Wang, Fang Dai, Li Zhang, Suixin Lu, “An Image Retrieval Method Based on Texture Features of Object Region”, *IEEE International Coriference on Electronics and Optoelectronics,* 2011.

[47] G Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval", *Pattern Recognition*, vol. 35, pp. 1675 – 1686, August, 2002.

[48] Histogram toolbox, http://www.viplab.cs.nott.ac.uk/download

[49] Vinay Kumar B, “A Thesis on Face Recognition Using Gabor Wavelets”, *Visvesvaray echnological University, Belgaum,* 2006.

[50] T. Ebrahimi, T.R. Reed, and M. Kunt, “Sequence coding by Gabor decomposition”, *In J. Torres, E. Masgrau, and M.A. Lagunas, editors, Signal Processing V: Theories and Applications, Proceedings of EUSIPCO-90,* pp. 769-772, 1990.

[51] Mian Zhou, “Facial Feature Extraction and Selection by Gabor Wavelets and Boosting”, *IEEE CISP '09, 2nd International Congress on Image and Signal Processing,* 2009.

[52] L. Wiskott, J.M. Fellous, N. Kruger, and C. Malsburg, “Face recognition by elastic bunch graph matching”, *In L.C. Jain et al, editor, Intelligent Biometric Techniques in Fingerprint and Face Recognition,* pp. 355–396,1999.

[53] Mian Zhou and Hong Wei, “Gabor-Boosting Face Recognition”, *Thesis at det.tjfsu.edu.cn/faculty/Thesis.pdf*, March 13, 2009.

[54] Henry Schneiderman , “A Statistical Approach to 3D Object Detection Applied to Faces and Cars”, *Thesis at csce.uark.edu/~jgauch/library/Other/Schneiderman.2000.pdf*, 2000.

[55] Subash Chelliah,” A Study of Euclidean classifier”, ACRS 1989, http://www.a-a-r-s.org/acrs/proceeding /ACRS1989/Papers/PS289-1.htm

[56] Euclidean distance, http://en.wikipedia.org/wiki/Euclidean_distance

[57] Naive-bayes-classifier, http://www.statsoft.com/textbook/naive-bayes-classifier/

[58] Naive-bayes-classifier, http://en.wikipedia.org/wiki/Naive_Bayes_classifier

[59] K-nearest_neighbor, http://www.scholarpedia.org/article/K-nearest_neighbor

[60] K-nearest_neighbor algo, http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[61] Boser, B. E., Guyon, I. M., & Vapnik, V., “A training algorithm for optimal margin classifiers”, *In Proc. of the 5th Annual Workshop on Computational Learning Theory, Pittsburgh,* pp. 144-152, 1992.

[62] M.F. Aka y, "Support vector machines combined with feature selection for breast cancer diagnosis", *Expert Systems with Applications 36,* pp. 3240-3247, 2009.

[63] Burges C, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery,* vol. 2, pp.121-127, Feb. 1998.

[64] Support vector machine, http://en.wikipedia.org/wiki/Support_vector_machine

[65] Neural Network, http://uhaweb.hartford.edu/compsci/neural-networks-history.html

[66] Silvia Cateni, Valentina Colla, Marco Vannucci and Alice Borselli, “Fuzzy Inference Systems Applied to Image Classification in the Industrial Field”, *Fuzzy Inference System - Theory and Applications,* 2009.

[67] Zadeh L.A., “Fuzzy sets. Information and Control”, *International Journal of Applied Math and Computer Science,* vol. 8, pp.338-353, 1965.

[68] hseker, “Fuzzy Membership Functions”, IMAT3406 Fuzzy Logic and Knowledge-based Systems, CSCI 3006.ppt .

[69] Kuncheva L.I., "Fuzzy vs Non-fuzzy in combining classifiers designed by boosting”, *IEEE Transactions on Fuzzy Systems,* 11 (6), pp. 729-741, 2003.

[70] Fuzzy classifiers, http://www.scholarpedia.org/article/Fuzzy_classifiers

[71] P. K. Sahoo, S. Soltani and A. K. C. Wong and Y. C. Chen, "A survey of thresholding techniques", *Computer Vision, Graphics, and Image Processing,* vol. 41, pp.233 – 260, 1988.

[72] Nita M. Nimbarte, Prof. Milind M. Mushrif, “Multi-level Thresholding Algorithm For Color Image Segmentation”, *IEEE Second International Conference on Computer Engineering and Applications,* pp.231-233, 2010.

[73] Robert Fergus, “Visual Object Category Recognition”, *Proc. of the IEEE Conf on Computer Vision and Pattern Recognition,* 2005.

[74] Mahboob H. Chowdhury and Warren D. Little, “Image Thresholding Techniques”, *Proc. of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing,* 1995.

[75] Hui-Fuang Ng, “Automatic Thresholding for Defect Detection”, *IEEE Proceedings of the Third International Conference on Image and Graphics (ICIG’04)*, 2004.

[76] Mansuo Zhao and Hong Yan, “Adaptive Thresholding Method For Binarization Blueprint Images”, *Fifth International Symposium on Signal Processing and its Applications, ISSPA ‘99, Brisbane, Australia, 22-25,* August 1999.

[77] H. Tian, S. K. Lam, T. Srikanthan, “Implementing Otsu’s Thresholding Process Using Area-Time Efficient Logarithm Approximation Unit”, *Proceedings of ISCAS '03 IEEE International Symposium on Circuits and Systems,* 2003.

[78] Liu Jianzhuang, Li Wenqing, and Tian Yupeng, “Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Otsu Method”, *China 1991 International Conference on Circuits and Systems,* June 1991.

[79] DongjuLiu, JianYu, “Otsu method and K-means”, *IEEE Ninth International Conference on Hybrid Intelligent Systems,* 2009.

[80] Fridtjof Johannes Stein, “Structural Indexing For Object Recognition”, *Thesis present at iris.usc.edu/outlines/papers/1992/stein-thesis-92.pdf,* 1992.

[81] Daniel Martin Keysers, "Approaches to Invariant Image Object Recognition", *Dissertation at www-i6.informatik.rwth-aachen.de/~keysers/da.pdf,* 2000.

[82] D.L. Swet and J. Weng, "Using discriminant eigenfeatures for image retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 831–836, 1996.

[83] Thomas Deselaers, "Features for Image Retrieval", *Dissertation* p*resent at www-i6.informatik.rwth-aachen.de/~deselaers/publications,* 2003.

[84] Scott Helmer, "Object Recognition with Many Local Features", *Thesis available at www.cs.ubc.ca/grads/resources/thesis/Nov04/Scott_Helmer.pdf,* 2004.

[85] Peter M. Roth and Martin Winter, "Survey of Appearance-Based Methods for Object Recognition", *survey available at www.icg.tu-graz.ac.at,* 2008.

[86] Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko, "Combining Local and Global Image Features for Object Class Recognition", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops, CVPR Workshops,* 2005.

[87] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *IJCV,* 2004.

[88] Ryszard S. Choras, "Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems", *International Journal of Biology and Biomedical Engineering,* issue 1, vol.1, 2007

[89] Jiri Matas and Stepan Obdrzalek, "Object Recognition Methods Based On Transformation Covariant Features", *In EUSIPCO,* 2004.

[90] B. J. Lei, Emile A. Hendriks, M.J.T. Reinders, "On Feature Extraction from Images", *MCCWS project Information and Communication Theory Group,* 1999.

[91] Vasileios Zografos, "Pose-invariant, model-based object recognition, using linear combination of views and Bayesian statistics", *Thesis at users.isy.liu.se/cvl/zografos/ publications/Thesis_FINAL.pdf,* 2009.

[92] B. Schiele, "Object Recognition using Multidimensional Receptive Field Histograms", *In 5th International Symposium on Intelligent Robotic Systems'97, Sweden* July 1997.

[93] Randal C. Nelson, "Object Recognition Research", *Research present at www.cs. rochester.edu/u/nelson/research/recognition/recognition.html.*

[94] Timo Ojala, Matti Pietikinen, and Topi Menp, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* pp.971–987, 2002.

[95] Krystian Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* pp. 1615–1630, 2005.

[96] M. Turk and A. Pentland, "Eigenfaces for recognition", *Journey of Cognitive Neuroscience,* pp. 71–86, 1991.

[97] M.S. Bartlett, H.M. Lades, and T.J. Sejnowski, "Independent component representations for face recognition", *Proceedings of SPIE,* pp. 528–539, 1998.