# UNIVERSITY OF NAIROBI

## SCHOOL OF ENGINEERING

# A BAYESIAN APPROACH TO AIRCRAFT IDENTIFICATION BY OUTLINE SHAPE USING INVARIANT IMAGE MOMENTS

By

Wambaa Dickson Gichaga

F56/76676/2009

A thesis submitted in partial fulfillment of the Degree of Master of Science in Electrical and Electronic Engineering in the Department of Electrical and Information Engineering in the University of Nairobi

July 2015

# DECLARATION OF ORIGINALITY FORM

This form must be completed and signed for all works submitted to the University for examination.

| | |
|---|---|
| **Name of Student:** | Wambaa Dickson Gichaga |
| **Registration Number:** | F56/76676/2009 |
| **College:** | College of Architecture and Engineering |
| **Faculty/School/Institute:** | School of Engineering |
| **Department:** | Department of Electrical and Information Engineering |
| **Course Name:** | Electrical and Electronic Engineering |
| **Title of the work:** | A Bayesian Approach to Aircraft Image Identification by Outline Shape Using Invariant Image Moments. |

## DECLARATION

1. I understand what Plagiarism is and I am aware of the University's policy in this regard.
2. I declare that this thesis is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.
3. I have not sought or used the services of any professional agencies to produce this work.
4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University Plagiarism Policy.

Signature

Date 01/02/2015

## DECLARATION

This research project is my original work and has not been submitted for an award of a degree in any University.

Signed_____ Date_____

Wambaa Dickson Gichaga

F56/76676/2009

This research project has been submitted for an examination with my approval as university supervisor.

Signed_____ Date_____

Prof Elijah Mwangi

Supervisor

University of Nairobi

## APPROVAL

This thesis has been submitted for examination with my approval as university supervisor.

Professor Elijah Mwangi

Signature……

Date……

# ACKNOWLEDGEMENT

I would like to thank my supervisor Prof. Elijah Mwangi for his immense support and for providing an interesting and motivating environment for research. I will never forget the day three years ago I asked Prof. Mwangi where optimization was utilized. Optimization has probably turned out to be the most important philosophy in my entire fields of work and not only for the purpose of this thesis. I have also to thank my colleagues who in the last few years shared the challenging activities related to research in particular Felix Owalla and Henry Kiragu.

I would also like to thank the Chairman of Department, Prof. H.A. Ouma and the head of technical section Mr. B.K Chomba for their encouragement and support. I would also like to express my gratitude to Prof. V. Oduol and Prof. J.M Mbuthia who introduced me to research methodologies.

Lastly but not least I thank my wife and my child. The completion of my thesis was only possible with their full support and understanding.

Wambaa Dickson Gichaga

## ABSTRACT

The outline shape is one of the most important attributes in the classification or identification of an object. It is used in human visual perception although the exact processes are yet to be understood. In computer vision system, the object shape is modelled as a mathematical function and characteristic features are then extracted. The features are selected on the basis of invariance to object orientation, rescaling and translation.

In this thesis, a set of image moment invariants have been used to represent the outline shape of military aircrafts. The Hu invariant moments have been used due to relative computational ease compared to other equivalent methods and their ability in preserving rotation, scaling and translation invariance.

Aircraft satellite images are identified using Bayesian decision theory classification. The classification is based on the statistical properties of a training set. The resulting statistical classifier assigns a measurement to the class which most likely generated the measurement. The classification is supervised for the sake of reduction of identification error rate. The accuracy of the proposed method has been tested by computer simulation experiments using the MATLAB R2009a version. The effect of noise can mask a shape descriptor and render the characteristic feature to be un-usable. The presence of speckle noise in satellite imagery has such effects. In our investigation various satellite images with different levels of noise are filtered using Lee filter, for the extraction of features and the results assessed. The Peak Signal-to-noise Ratio (PSNR) is used to assess the quality of the filtered images. The computer simulation results show that the proposed algorithms are effective in filtering, extraction of features and in the recognition of images. The test images used include 100X100 up to 1024X1024 gray level images.

## TABLE OF CONTENTS page

**DECLARATION ……..ii**

**APPROVAL …… iii**

**ACKNOWLEDGEMENTS ……iv**

**ABSTRACT ……..v**

**TABLE OF CONTENTS ……vi**

**ABBREVIATIONS AND ACRONYMS ……..x**

**LIST OF FIGURES ……xi**

**LIST OF TABLES ……..xii**

**CHAPTER 1** **1**
INTRODUCTION ……1

**1.1 Background to the Study ……1**

**1.2 Problem Statement ……...8**

**1.3 Objectives ……..……9**

**1.4 Scope ……...9**

**1.5 Thesis Organization ……..……...9**

**CHAPTER 2** 10
LITERATURE REVIEW ……10

**2.1 Shape features Properties ……10**

**2.2 Invariant Moments in Pattern recognition Techniques ….11**

**2.3 Bayesian Techniques in Pattern recognition ……13**

**2.4 Vector Quantization Techniques ……..……14**

**2.5 Pattern Recognition Techniques in medicine ……..……15**

**2.6 Aircraft pattern Recognition ……..……16**

**2.7 Proposed Method ……...18**

**CHAPTER 3 28**

IMAGE PATTERN FEATURES ……..…….28

**3.1 Shape Parameters ……..….28**

**3.2 Shape Signatures ……...29**

**3.3 Chain Code ……..…….……32**

**3.4 Moments ……34**

**3.5 Shape Transform Domain ……..…41**

**CHAPTER 4 47**

BAYESIAN DECISION BASED IMAGE RECOGNITION …...47

**4.1 Bayes Classifier ……...47**

**4.2 Approaches to Statistical Pattern Recognition ……...49**

**4.3 Bayes Decision Rule for Minimum Error ……...51**

**4.4 Vector Quantization ……54**

**CHAPTER 5**

RESULTS AND DISCUSSIONS …….60

**5.1 Preprocessing ……..60**

**5.2 Filter performance ……...65**

**5.3 Invariant Image Moments ……66**

**5.4 Bayesian Classification ……76**

**5.5 Vector Quantization ……...81**

**5.6 Optimization ……86**

**CHAPTER 6** 99

CONCLUSIONS AND RECOMMENDATIONS ……... 99

**6.1 The Classifier ……99**

**6.2 Misrecognition ……100**

**6.3 Recommendations ……..101**

**REFERENCES ……..102**

APPENDIX ……..107

APPENDIX A : MATLAB CODE ……108

APPENDIX B: VECTOR QUANTIZATION ……..117

SYMPOSIUM ……..120

APPENDIX C: PUBLISHED PAPER ……121

# ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| DFT | Discrete Fourier Transform |
| DWT | Discrete Wavelet Transform |
| FMT | Fourier-Mellin Transform |
| FT | Fourier Transform |
| LPM | Log-Polar Mapping |
| LSB | Least Significant Bit |
| MSE | Mean Square Error |
| MT | Mellin Transform |
| NC | Normalized Cross-correlation |
| NTSC | National Television System Committee |
| PSNR | Peak Signal-to-Noise Ratio |
| RF | Rotation Factor |
| RST | Rotation, Scaling and Translation |
| SF | Scaling Factor |
| TF | Translation Factor |
| VQ | Vector Quantization |

# LIST OF FIGURES


Figure 2.1 Steps in image recognition ……..19

Figure 2.2 Unprocessed satellite image ..……20

Figure 3.1 Chain code directions ……...32

Figure 3.2 Vertex chain code ……....33

Figure 3.3 Axis of least inertia…….34

Figure 3.4 Tangent angle ……42

Figure 3.5 Polar into Cartesian coordinate image ……..45

Figure 4.1 Posterior probabilities ..……53

Figure 4.2 Likelihood ratio and threshold …….54

Figure 4.3 Nearest neighbor classifier rule ……..57

Figure 4.4 Hypothetical placement of the codebook vectors ……58

Figure 4.5 An illustration of Voronoi regions ……59

Figure 5.1 Unprocessed images ……61

Figure 5.2 Typical pre-processed aircraft satellite images ……61

Figure 5.3 Images of B2 bomber with different noise levels …….64

Figure 5.4 A noisy and denoised image of B2 bomber ……..65

Figure 5.5 Invariance tests …….68

Figure 5.6 Data scatter figure ……....85

Figure 5.7 The F35 aircraft .....86

Figure 5.8 Five clusters silhouette …….88

Figure 5.9 Ten clusters silhouette ……..89

Figure 5.10 Leave one out method …… 95

Figure 5.11 Yak 7 and A7 Corsair II aircrafts…… 97

# LIST OF TABLES


Table 5.1: PSNR levels ……64

Table 5.2: Invariant moments of an image
With different levels of noise ……67

Table 5.3: Invariance tests ……..68

Table 5.4 Features extracted from fighter aircrafts ……70

Table 5.5 Features extracted from cargo aircrafts ……71

Table 5.6 Features extracted from bomber aircrafts ……..72

Table 5.7 Features extracted from rotor wing aircrafts ……..73

Table 5.8 Features extracted from reconnaissance aircrafts ……74

Table 5.9 Features extracted from training aircrafts ……..75

Table 5.10 Unknown images features ……77

Table 5.11a Aircraft type Numbers ……78

Table 5.11b Class prior probabilities ……79

Table 5.12 Fighter class/unknown aircraft I distances ……79

Table 5.13 Rotary wing/unknown aircraft II distances ……..80

Table 5.14 Invariant moments for training patterns ……..81

Table 5.15a k-means clustering ……83

Table 5.15b k-means clustering for unknown aircrafts …….. 84

Table 5.16 Clustering analysis ……86

Table 5.17 Distance minimization of 10 clusters ……...90

Table 5.18 Distance minimization of 5 clusters ……90

Table 5.19 Error Table ……...95

# CHAPTER 1

# INTRODUCTION

### 1.1. Background to the Study

As a scientific discipline, computer vision is concerned with the extraction of information from images to be employed in a decision making process. The image data can take many forms, such as video sequences, still images, digitized maps, diagrams and sketches. These images may be in colour format, grey scale or in binary format. The common approach is to extract characteristic features from the image either in the spatial domain or in some suitable transform domain. Whether the goal is classification or recognition, a measure of similarity or distance must be formulated and the success rate of the system evaluated. In all these applications the basic approach is the one of pattern recognition. It is based on extracting representative features from objects and then identifying measurable quantities that make these objects distinct from each other compared to objects in a given database. In order to extract the characteristic features from an image pattern several preprocessing procedures are necessary. These are:

i. Re-sampling in order to assure that the image coordinate system is correct.
ii. Noise reduction in order to assure that sensor noise does not introduce false information.
iii. Contrast enhancement to assure that relevant information can be detected.

Some of the application areas of computer vision are:

#### 1.1.1. Robotics

The visual recognition of image patterns is a fundamental human attribute that uses the eye as a sensor and dedicated parts of the brain as the decision making processor. The visual recognition enables humans to perform a variety of tasks such as target identification, ease of movement, handling tools, and communication among others. Advances in sensing and visual perception techniques have enabled some of these attributes to be transferred to robots. For example, identification of colour or shape are a useful asset in a robot in an industrial automation process that involves sorting of items [1], [2], [3].

### 1.1.2. Industrial Inspection

Industrial inspection is an area in which pattern recognition is of importance. A pattern recognition system captures images via a camera and analyzes them to produce descriptions of the visual signal. For example, objects on a moving conveyor belt may pass an inspection station under the surveillance of a camera to detect defective or faulty parts. Therefore, images have to be analyzed online, and the pattern recognition system has to classify the objects into either “defective” or “non-defective” class. After that, a decision has to be taken, such as to reject the defective parts.

In an assembly line, different objects must be classified in one of a number of classes known a priori. Examples are in a tools manufacturing plant, where a robot arm can be used to move objects in the right place [4], [5].

A well-established application of image pattern recognition is in assembly of electronic components on printed circuit boards in order to isolate defects such as shorts, opens, over-etching, under-etching, pad size violations, and spurious metals [6]. There is also need to identify typical solder joint defects on PCBs such as missing solder, cold solder, excess solder, blowholes, voids, and solder bridging. Mistakes of component insertion can also be inspected. A printed circuit board inspection system based on image processing and pattern recognition techniques has been developed and reported by Cao and Meng [6]. It involves a binary image drawn from the original PCB and comparing the areas or the circumferences of the two, evaluation can be carried out. Another technique is to compare the acquired image with the perfect one. The difference between the two images will influence the decision that will be made [6].

### 1.1.3. Biometric Identification and Verification

Threat to security has born new ways for protection of software, hardware, and network systems from external attacks. One of the methods is by using biometric systems. Such systems use personal attribution in an individual that are unalterable and hence popular as identification and verification features. Types of image based biometrics are fingerprint, iris, eye retina, hand, gait, and the face. The fingerprint is a typical pattern in the form of lines network referred to as ridges and minutiae. Everyone’s fingerprint pattern has a unique form even among identical twins. The analysis of fingerprints for matching purposes generally requires the comparison of several features of the fingerprint pattern. These include patterns, which are aggregate characteristics of ridges, and minutiae points, which are unique features

found within the patterns. It is also necessary to know the structure and properties of human skin in order to successfully employ some of the imaging technologies.

Pattern recognition algorithms compare the basic fingerprint patterns such as the arch, whorl, and loop between a previously stored template and a candidate fingerprint. This requires that the images be aligned in the same orientation. To do this, the algorithm finds a centroid in the fingerprint image and centers on that. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match [7].

Another type of biometric identification is iris recognition which exploits its unique pattern in an individual. Image templates encoded from pattern recognition algorithms allow the identification of an individual from an impostor as databases of enrolled templates are rapidly searched by pattern classifiers. These systems enable passport-free border-crossings, and some national ID systems based on this technology are being deployed. A key advantage of its recognition is its stability as an internal, protected, yet externally visible organ of the eye.

An algorithm first has to realize an accurate registration of the inner and outer boundaries of the iris in a captured image. Further processes detect and exclude eyelids, eyelashes, and specular reflections that often occlude its parts. The image region that contains only the iris, is then analyzed to extract a bit pattern encoding the information needed to compare two of its images. For identification or verification, a template created by its imaging is compared to a library of stored templates in a database. A similarity distance measure is used as a decision threshold. A positive identification is ensured because of the statistical extreme improbability of two different persons having similar patterns [8].

Another biometric technique is the retinal identification [6]. Retinal recognition is the most reliable and stable means of biometric identification. The retina is not exposed to the external environment. As a biometric, it is therefore very stable. The retina consists of multiple layers of sensory tissue and millions of photoreceptors whose function is to transform light rays into electrical impulses. These impulses subsequently travel to the brain via the optic nerve, where they are converted to images. It is the blood vessel pattern in the retina that forms the foundation for retinal recognition as a pattern recognition technique, even among identical twins - the blood vessel patterns of the retina are unique and different. Infrared light illuminates the blood vessel pattern of the retina and the reflected light is subsequently captured by a scanning device for processing. Consequently, algorithms are developed for the extraction of unique features.

An additional biometric is the facial recognition system which is a pattern recognition system for identification or verification of a person from a digital image or a video frame. This is done

by comparing selected facial features from an image and a facial database. It is typically used in security systems.

Facial recognition algorithms identify facial features by extracting features, from an image of the face. These features are then used to search for other images with matching features in a database. Other techniques normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. The recognition processes are mainly of the geometric type, which aims at distinguishing features, or photometric type, which is a statistical approach that compresses an image into luminance values and compares the values with stored templates to eliminate variances [2].

### 1.1.4. Medical Imaging

Computer-aided diagnosis is also an important application of pattern recognition, aiming at assisting health personnel in making diagnostic decisions. Computer-assisted diagnosis has been applied to and is of interest for a variety of medical data, such as X-rays, Computer tomography images, Ultrasound images, Electrocardiograms (ECGs), Magnetic resonance imaging (MRI), Positron emission tomography (PET) and electroencephalograms (EEGs). The need for a computer-aided diagnosis stems from the fact that medical data are often not easily interpretable, and the interpretation can depend very much on the skill of the medical analyst. However, a pattern recognition system increases the chances of correct diagnosis thus assisting the health personnel [5].

Data mining in medicine and DNA data analysis has also seen an increased usage recently. The nucleotides in a DNA chains have four basic building elements referred to as: adenine (A), cytosine (C), guanine (G) and thymine (T). These four nucleotides are combined to form long sequences in a twisted ladder structure commonly referred to as helix. Several nucleotides, usually in the range of hundreds, when arranged in a particular order constitute a gene. Specific gene-sequence patterns are related to particular diseases and play an important role in medicine and genetic engineering. In this, pattern recognition is a key area that offers a wealth of developed tools for similarity search and comparison between DNA sequences. Such comparisons between healthy and diseased tissues are very important in medicine to identify critical differences between these two classes [5]. Similarly the comparison is applied in DNA pattern identification of corpses that cannot be identified in any other way due to decomposition or destruction in fires or air accidents. To add to that DNA pattern recognition is applied in forensic investigations in crime where the DNA Patterns of the victim or criminal is identified.

### 1.1.5. Surveillance

Image recognition offers several opportunities for airborne surveillance for applications in civilian and military activities, such as search and rescue missions, border security, resource exploration, wildfire and oil spill detection, and target tracking, among others. An unmanned airborne vehicle is equipped with special sensors to image objects in ground and assigns the actual recognition task to a crew or record image data and analyze them off-line on the ground [9].

In the same arena, pattern recognition is applied in image-based automatic surveillance for airport surface. The advanced surface movement guidance and control systems require the unambiguous identification of airports runways and all aircraft and vehicles in the airport movement area, as well as accurate information on their position. A pattern recognition system provides users with a display of the location of all surface traffic, enabling its separation and guidance in all types of weather conditions without reducing the number of operations or the level of safety.

The automatic target recognition (ATR) as a process of automatic target acquisition and classification is extremely important for defense applications. The generic ATR problem is to take information from one or more sensors, and if necessary, combine it with a priori information. A decision is then made about the type of targets present in the scene. The targets are usually prioritized by their tactical importance so that appropriate actions can be taken in a given situation once their presence has been inferred. The identification of enemy military hardware and installations in fighter aircraft is an example of the employment of such technologies [10].

**1.1.6. Character Recognition** Recognition of letters or numbers is another important area of pattern recognition, with major concerns in automation and data mining. An optical Character Recognition (OCR) system has a “front-end” device consisting of a light source, a scan lens, a document transport, and a detector. At the output of the light-sensitive detector, light-intensity variation is translated into “numbers” and an image array is formed. A pattern recognition algorithm then takes over to recognize the characters—that is, to classify each character in the correct “letter, number, punctuation” class. Storing the recognized document has a twofold advantage over storing its scanned image. First, further electronic processing, if needed, is easy via a word processor, and second, it is much more efficient to store ASCII characters than a document image. There is also a lot of interest in systems that recognize handwriting. A typical commercial application of such a system is in the machine reading of bank cheques. The machine must be able to recognize the amounts in figures and digits and match them. Furthermore, it may also check whether the payee corresponds to the account to be credited. Online handwriting pattern

recognition systems are another area of great commercial interest. Such systems accompany pen computers, with which the entry of data is done not via the keyboard but by writing [11], [12], and [13].

**1.1.7. Content based image retrieval** Content based image retrieval is the application of pattern recognition techniques in searching for digital images in large databases. It uses the visual contents of an image such as colour, shape, texture, and spatial layout to represent and index an image. In a typical system, the visual content of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors. The similarities or distances between the feature vectors of the query or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. The indexing scheme provides an efficient way to search for the image database [6].

## 1.2. Problem Statement

The need to recognize aircraft type from shape arises in content based image retrieval and in surveillance. A pattern recognition process can be employed for identification of an input aircraft image by searching a library database. A successful system would need to extract feature patterns from the image that would be invariant to rotation, scaling, and translation. It would be expected that features from similar patterns would show some statistical deviations as they are captured under a different environment such as noise levels and measurement errors. Thus a decision making process that takes into consideration any prior knowledge of the patterns and also uses a statistical decision would provide a suitable solution to the problem.

## 1.3. Objectives

The main objective of this investigation is to identify aircrafts from satellite images that are stored in a database. The specific objectives are:

(i). To use preprocessing techniques such as noise reduction in order to enhance the image of the unknown aircraft.

(ii). To extract characteristic features such as moments which are resistant to image geometric rotations, translations and scaling.

(iii). To employ Bayesian decision making in the image recognition process.

### 1.4. Scope of the Work.

The research work is limited to identification of military aircrafts from aerial view images taken by satellite. The recognition of aircraft types from structural diagrams and sketches is outside the scope of this thesis.

### 1.5 . Thesis Organization

The remaining part of the thesis is organized as follows. In chapter 2, the literature review that reports on various pattern recognition techniques that have been proposed by other researchers is presented. Chapter 3 covers diverse image features extraction methods but with a focus on moment invariant descriptors. Chapter 4 covers Bayesian decision based image recognition process while chapter 5 presents computer simulation results and the discussion. Lastly, in chapter 6, the conclusions and recommendations for further work are presented.

# CHAPTER 2

# LITERATURE REVIEW

This chapter gives a brief review of the techniques that have been proposed or used in pattern recognition. It reviews decision theoretic approaches used in the proposed technique such as moment invariants and Bayesian decision method. Pattern recognition is concerned primarily with the description and classification of measurements taken from physical processes. Approaches to computerized pattern recognition may be divided into two principle areas namely decision theoretic or structural methods.

In the decision theoretic approach the patterns are described using quantitative descriptors taken from the spatial domain such as area, contour and beam angle, In addition, transform based descriptors such as moments, Fourier coefficients, Wavelets coefficients, can also be used. In the structural approach the patterns are described by symbolic information such as strings and the relationships and properties between those symbols.

## 2.1. Shape Features Properties

Efficient shape features must present some essential properties such as:

**i. Characteristic features: This means that shapes that are perceived to be similar must possess features that distinguish them from dissimilar types. This represents compactness of the representation. In general, a shape descriptor is a set of vectors that are produced to represent a given shape feature. A descriptor attempts to quantify the shape in ways that agree with human intuition. Good retrieval accuracy requires a shape descriptor to effectively find perceptually similar shapes from a database.**

**ii. Geometric invariance: This means that the location, the rotation and the scaling changing of the shape must not affect the extracted features. The affine transform performs a linear mapping from coordinates system to other coordinates system that preserves the "straightness" and "parallelism" of lines. The transform can be constructed using sequences of translations, scales, flips, rotations and shears. The extracted features must be as invariant as possible with the affine transform. When some parts of a shape are occulted by other objects, the feature of the remaining part must not change compared to the original shape.**

**iii. Noise resistance: Features must be as robust as possible against noise, this means that features must not be changed significantly by noise levels. In order to satisfy this**

**requirement, efficient noise renewal methods must be incorporated as a preliminary stop in pattern processing.**

**iv. Completeness and Compactness: The descriptors should be as complete as possible in representing the content of the information items. Also the size of a descriptor vector must be compact while the computation of the similarity or the distance between descriptors should be simple so as to reduce the execution time.**

### 2.2. Invariant Moments in Pattern recognition Techniques

A. Kadir et al [14] have proposed an approach to identify plants using leaf shapes. A combination of geometric features such as moment invariants were used as characteristic features. In this technique, a comparative experiment of four methods to identify plants using leaf shape features was accomplished. In their work, Zernike and Polar Fourier transforms were employed for the first time in leaf identification.

The moment invariants method has been widely used and has been developed to describe shape irrespective of position, orientation and scale.

L. Keyes and A.C. Winstanley [15] have presented feature coding and recognition of topographic data. The topographic data was derived from air survey or raster scanning large-scale paper maps, requiring classification of objects such as buildings, roads, rivers, fields and railways. The applicability of the above method to topographic shapes in the paper is described and its usefulness evaluated. The method derives descriptors consisting of a small number of real values from the object's polygonal boundary. Two large corpora representing data sets from ordnance survey maps were used. The effectiveness of the description technique was evaluated by using one corpus as a training-set to derive distributions for the values for supervised learning. This was then used to re-classify the objects in both data sets using the descriptor to evaluate its effectiveness.

J. Flusser has presented a survey of object recognition and classification methods based on image moments [16]. Various types of moments (geometric moments, complex moments) and moment-based invariants with respect to various image degradations and distortions (rotation, scaling, affine transform, image blurring, etc.) which can be used as shape descriptors for classification are reviewed. The paper also presents a general theory on the construction of moment invariants and show also a few of them in explicit forms. Efficient

numerical algorithms that can be used for moment computation are reviewed and demonstrated practically.

L. Kotoulas and I. Andreadis [17], present an overview of the most commonly used image moments, namely geometric, complex, Zernike, Tchebyshev and Hahn. A hardware architecture capable of fast calculation of geometric, Zernike and Tchebyshev moments is also presented.

Jorge Leon-Garcia et al [18] propose an automatic fingerprint recognition system based in fingerprint features and invariant moments. In this work the enhanced image is processed using two algorithms, the first one is Fast Fourier Transform (FFT) and the second one is a bank of Gabor filters. After the enhancement process, an algorithm to extract the minutiae information is applied obtaining distance, angle and coordinates. The invariant moments are used to discriminate between those fingerprints that are confused.

A. Halet et al [19] propose a classification of images affected by different levels of noise. This classification is carried out for images of different texture, such as human faces, aircraft images, and binary images. Generic moment invariants are introduced that increase the identification chances without affecting invariance properties.

G. Dimitoglou et al [20] propose a technique of facial expression recognition with the help of several properties associated with the face itself. As facial expression changes, the curvatures on the face and properties of the objects such as eyebrows, nose, lips and mouth area also changes. Hu moment invariants have been used to compute these changes and results are recorded as feature vectors. An artificial neural network was used as a classification tool. The generalized feedforward neural network recognizes seven universal expressions i.e. anger, disgust, fear, happiness, sadness, and excitement as well as gestures. The neural network was trained and tested by using the scaled conjugate gradient back propagation algorithm.

### 2.3. Bayesian Techniques in Pattern recognition

M. Husnain and S. Naweed [21] use a Bayesian technique to identify each of the large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within 20 fonts was randomly distorted to produce a file of 20,000 unique instances. The features of the dataset and the errors committed by Holland-style adaptive classifiers were

analyzed in an attempt to use Bayesian decision theory in-order to reduce the error rate. At the end, principal component analysis is applied for dimensionality reduction.

K. Jayech et al [22] propose a new approach based on enhanced naive Bayes theorem for image classification. For each block of images, a vector of descriptors is computed. Then, a classification of the vectors of descriptors to build a vector of labels for each image is carried out. Finally, three variants of Bayesian networks are proposed. The results showed a marked improvement over the normal networks.

**2.4. Vector Quantization Techniques**

A. Giusto [23] gives an introduction to the class of algorithms proposed by Kohonen (2001) [24] known as Learning Vector Quantization (LVQ) whose purpose is to classify data into a discrete and finite set of classes. A VQ classifier is based on the definition of certain key points referred to as codebook vectors in the data space. Once these codebooks are singled out, the new data is classified to belong to the same class of the closest codebook vector in the Euclidean metric. This highlights the key difference between a Bayesian and a VQ classifier; while the Bayesian strategy seeks to approximate the discriminant functions over the whole space used to represent the data, the VQ classifier focuses on a small region of this space, where most of the action takes place.

S. Thepade et al [25] propose to reduce response time, computation complexity and search space while classifying an image using vector quantization. In this research paper, a novel technique based on vector quantization for fingerprint classification using Linde, Buzo, and Gray (LBG) also called Generalized Lloyd Algorithm (GLA) is proposed. The classification is done on fingerprint images using LBG codebooks of sizes 4. The proposed technique takes lesser computations as compared to usual fingerprint classification techniques.

A. Witoelar et al [26] propose Learning Vector Quantization (LVQ) a popular multi-class classification algorithm. Prototypes in an LVQ system represent the typical features of classes in the data. Multiple prototypes are employed for a class to improve the representation of variations within the class and the generalization ability. In this paper, the dynamics of LVQ in an exact mathematical way, aiming at understanding the influence of the number of prototypes and their assignment to classes is investigated. Using a system of three prototypes the different behaviours of LVQ systems of multiple prototype and single prototype class representation is demonstrated.

### 2.5. Pattern Recognition Techniques in medicine

I. Sommer et al [27] propose an approach for identifying similarities of protein–protein binding sites. The geometric shape of a binding site is described by computing a feature vector based on moment invariants. In order to search for similarities, feature vectors of binding sites are compared. Similar feature vectors indicate binding sites with similar shapes.

A classification of lung nodules in low dose CT scans using geometric feature description has been presented by A. Farag et al [28]. In the method, the paper examines the effectiveness of geometric feature descriptors, common in computer vision, for false positive reduction and for classification of lung lumps in low dose CT scans. A data driven lung nodule modeling approach creates templates for common lump types, using active appearance models, which are then used to detect candidate lumps based on optimum similarity measured by normalized cross-correlation. Geometric feature descriptors are applied to the output of the detection step, in order to extract features from the nodule candidates, for further enhancement of output and possible reduction of false positives. Thus quantitative measures of enhancements of the performance of CAD models based on low dosage CT are now possible and are entirely model-based.

P. D. Avani et al [29] propose an algorithm to identify the type of malaria causing parasite through their colour and shape. Malaria is a serious global health problem, and rapid, accurate diagnosis is required to control the disease. An image processing algorithm to automate the diagnosis of malaria in blood images is reported in this paper. In blood samples, if the red corpuscles of human beings are infected by malarial parasites, they will have a specific shape which can identify their presence. The research suggests that the shape of the affected red blood cells can be detected using 2D moments of the image of the infected cell.

A medical manufacturing corporation called Welcome, working at the University of California, Berkeley, displayed an image on the internet showing two red blood cells taken from a patient with sickle cell disease. The cells were imaged by scanning electron microscopy. Two red blood cells, one normal and one diseased, were then processed using pattern recognition techniques to highlight the striking differences in their shapes and thus enabling the detection of anaemia [30].

### 2.6. Aircraft pattern Recognition

J.W. Hsieh et al [31] propose image processing techniques to perform image preprocessing tasks, such as image quality enhancement, noise removal, automatic binarization, and rotation, scaling, translation adjustment. Then, distinguishable shape features derived from the characteristics exhibited by aircrafts are extracted on which the recognition is based. Last, multi-level recognition scheme is adopted to recognize the types of aircrafts by incorporating suitable weight into each recognition level. Experimental results reveal the feasibility and validity of the proposed approach in recognizing aircrafts in satellite images.

W. Li et al [32] propose an airplane detection approach based on visual saliency computation and symmetry detection. The advantages are two fold. First, saliency and symmetry detection perform in a stable way in obtaining target location and orientation information. Second, independent of target type, pose and size, saliency map and symmetry detection are computed only once. This saves a large amount of computational time but does not miss any targets. Experiments show that the method provides a promising way to detect airplanes in complex airport scenes.

E. Polat et al [33] propose detection and recognition of stationary aircrafts in airports. In the study, a learning-based system that detects stationary aircrafts in satellite images obtained from Google Earth is developed. The features that emphasize the geometric structure of an aircraft are determined using a 2D Gabor filter. The aircraft detection is performed using Support Vector Machines (SVM) classification method. The SVM is a supervised learning method that analyzes data and recognizes patterns for classification. It takes a set of input data (a vector consists of Gabor filter output of images) and predicts the one of two classes (aircraft or non-aircraft). The performance of the system is demonstrated using satellite images collected from airports in Europe and United States.

K. Roopa et al [34] describe an aircraft recognition system implemented using moment invariant technique and discrete cosine transform. They provide a performance comparison between the two methods by using the static images from a database of images of aircraft and the captured image of the test aircraft.

F.W. Smith et al [35] undertook to study the feasibility of automatic interpretation of ship photographs using the spatial moments of the image as features to characterize the image. The photo interpretation consisted of estimating the location, orientation, dimensions, and heading of the ship. The study used simulated ship images in which the outline of the ship was transformed into a binary image to give a low-resolution high-contrast image of the ship

such as might be obtained using high resolution radar. The estimates were made using polynomials of invariant moments formed by transformations of the original spatial moments; density-invariant moments, central moments, and rotation-invariant moments. The transformations to invariant moments were chosen using linear regression. The best moments for the polynomials were chosen using linear regression.

### 2.7. Proposed Method

#### 2.7.1. Stages in the proposed pattern recognition method

Figure 2.1 is an illustration of a four stage process that shows the steps in image recognition. The presentation is grossly oversimplified. The data may undergo several separate transformation stages before a final outcome is reached. These transformations operate on the data in a way that usually reduces its dimension (reduces the number of features), removing redundant or irrelevant information, and transforms it to a form more appropriate for subsequent classification.

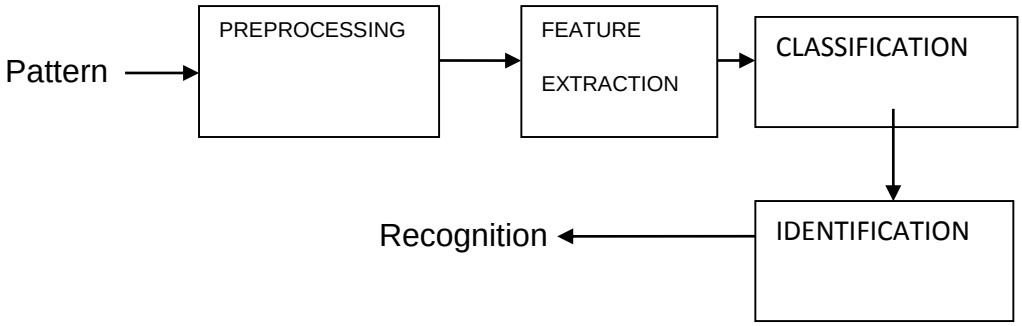



*Fig 2.1 Steps in Image Recognition.*

#### 2.7.2. Preprocessing

One of the main tasks in preprocessing an image is the removal or suppression of noise. There are two basic approaches to image denoising, spatial filtering methods and transform domain filtering methods. A traditional way to remove noise from image data is to employ spatial filters. Spatial filters can be further classified into non-linear and linear filters. The transform domain filtering methods can be subdivided according to the choice of the base functions. The basis functions can be further classified as data adaptive and non-adaptive. One of the popular transform domain filtering methods is the use of wavelets. In such

operations, the filtering process can be can be subdivided into linear and nonlinear. Linear filters such as the Wiener filter in the wavelet domain yield optimal results when the signal corruption can be modeled as a Gaussian process and the accuracy criterion is the Mean Square Error (MSE). Depending on the noise types, the noisy image can be generally modeled as one of the two models: additive model or multiplicative model. Speckle noise is a form of multiplicative noise which is more difficult to suppress than additive noise, it obscures small and low-intensity features. Many images are acquired under less than ideal conditions and consequently are contaminated by significant amounts of noise. Due to noise in the images, their segmentation, classification and detection becomes difficult. It is important for noise suppression algorithms to reduce the noise while preserving the important features in the images. Figure 2.2 illustrates a typical unprocessed satellite image. It can be noted that the object in the figure is barely recognizable due to the blurring effects of noise.

*Fig 2.2 Unprocessed satellite image [33]*

One major characteristic of images is their non-stationary nature and have many local features which are not very well described by using global features. Therefore, performance of linear algorithms used for image processing is limited. Adaptive nonlinear techniques are required for better performance. Linear techniques can cause blurring of edges whereas nonlinear techniques are edge preserving but still most of these techniques suffer from drawbacks like sensitivity to the size and shape of the filter window, in addition most of these techniques do not enhance the edges, and may not suppress noise in the neighbourhood of the edges. The hard thresholds used in these techniques may result in artifacts in the smoothed output images. In our proposed method, the speckle noise is reduced by using the Lee filter. The filter and other related adaptive filters essentially smooth homogeneous regions by averaging and avoid smoothing where edges may exist. The presence or absence of edges is based upon local coefficient of variation – higher values in the regional coefficient of variation indicate the location of edges. [36]

### 2.7.3. Denoising

The Lee filter is an adaptive filter which changes its characteristics according to the local statistics in the neighbourhood of the current pixel. It is able to smooth away noise in flat regions, but leaves the fine details (such as lines and textures) unchanged. The Lee filter is designed to eliminate speckle noise while preserving edges and point features in satellite images. Based on a linear speckle noise model and the Minimum Mean Square Error (MMSE) design approach, the filter produces the enhanced data. It uses a small window (3×3, 5×5, and 7×7) in which the local mean and variances are estimated. The distinct characteristic of the filter is that in the areas of low signal activity (flat regions) the estimated pixel approaches the local mean, whereas in the areas of high signal activity (edge areas) the estimated pixel favours the corrupted image pixel, thus retaining the edge information [36]. It is generally claimed that human vision is more sensitive to noise in a flat area than in an edge area [36]. The major drawback of the filter is that it leaves noise in the vicinity of edges and lines. However, it is still desirable to reduce noise in the edge area without sacrificing the edge sharpness.

The Lee filter modifies the image based on statistics extracted from the local environment of each pixel. It varies the contrast stretch for each pixel depending upon the Intensity values in the surrounding moving kernel. Obviously, a filter that adapts the stretch to the region of interest (the area within the moving kernel) would produce a better enhancement.

The basic equations of Lee filter as proposed by Jun-Wei Hsieh et al [31] are given in equation 2.1.

$$Intensity\ value_{out} = [Mean] + K[Intensity_{in} - Mean] \quad (2.1)$$

Where Mean= average of pixels in a moving window

Intensity value is the intensity of a particular pixel.

$$K = \frac{Var(x)}{[Mean]^2\ \sigma^2 + Var(x)} \quad (2.2)$$

And

$$Var(x) = \left(\frac{[Variance\ within\ window] + Mean\ within\ window^2}{Sigma^2 + 1}\right) - [Mean\ within\ window^2] \quad (2.3)$$

Where Mean and σ are equal to the local mean and variance respectively of all pixels within the user-selected moving kernel.

### 2.7.3. Features Extraction

A major problem associated with feature extraction is dimensionality. The number of features at the disposal of the designer of a classification system is usually very large. The number of features should be reduced to a sufficient minimum. This is to avoid computational complexity. Another major reason is the ratio of the number of training patterns N to the number of free classifier parameters, which lead to generalization properties of the resulting classifier. The basic questions arising in a classification task can be outlined as follows;

i. **How are the features generated? It is problem dependent, and it concerns the feature generation stage of the design of a classification system that performs a given pattern recognition task.**
ii. **What is the best number N features to use? This is also a very important task and it concerns the feature selection stage of the classification system. In practice, a larger than necessary number of feature candidates is generated, and then the "best" of them are adopted.**

Invariant moments will be used to form feature vectors of dimension 7, this figure arises from the Hu's formulas for invariant moments, as proposed in techniques used by Mingqiang et al [37], M. Hu [1], Dudani et al [10], Keyes L.et al [15], A. Farag et al [28], A. Kadir et al [14], W. Li et al [32], J. Kim [38], A. Halet et al [19], and K. Roopa, [34]

### 2.7.4. Classification

i. Having adopted the appropriate features, how does one design the classifier for the specific task? The line should be drawn optimally, with respect to an optimality criterion. In general, the surfaces dividing the space in the various class regions are nonlinear. What type of nonlinearity must one adopt, and what type of optimizing criterion must be used in order to locate a surface in the right place in the N-dimensional feature space? These questions concern the classifier design stage.
ii. Finally, once the classifier has been designed, how can one assess the performance of the designed classifier? That is, what is the classification error rate? This is the task of the system evaluation stage.

The Bayesian decision approach will be employed. It has several theoretical advantages including the property that the rate of miss-classification is minimized. For image processing simulation MATLAB tools will be utilized. These tools are selected because they are easily available, are not expensive and are easy to use. Boundaries of geometric objects will be

created with different rotation, translation, scale and shape and then classified using a database of images with known object types.

**2.7.4.1 Unsupervised Classification** In unsupervised learning or clustering the system forms clusters of the input patterns. Given a particular set of patterns or cost function, different clustering algorithms lead to different clusters.

The *k*-means clustering is a partitioning method. It partitions data into *k* mutually exclusive clusters, and returns the index of the cluster to which it has assigned each observation. Unlike hierarchical clustering, *k*-means clustering operates on actual observations (rather than the larger set of dissimilarity measures), and creates a single level of clusters. The distinctions indicate that *k*-means clustering is often more suitable than hierarchical clustering for large amounts of data. The *k*-means algorithm treats each observation in the data as an object having a location in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. One can choose from five different distance measures, depending on the kind of data that is to be clustered. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized. The *k*-means algorithm computes cluster centroids differently for each distance measure, to minimize the sum with respect to the measure that one specifies. It uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well-separated as possible. The details of the minimization using several optional input parameters to *k*-means, including ones for the initial values of the cluster centroids, and for the maximum number of iterations can be controlled.

**2.7.4.2. Supervised Classification** In supervised pattern recognition a set of training data are available, and the classifier is designed by exploiting this a priori known information. This is also known as machine learning.
Bayes classification requires only a small amount of training data to estimate the parameters (means and variances of the variables necessary for classification). Variables are assumed to be independent, only the variances of the variables for each class need to be determined and not the entire covariance matrix. All model parameters are approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. The class prior is calculated by assuming equiprobable classes [21], [22].

I.e. prior=1/number of classes. 2.4

Feature vectors will be treated as random vectors. This is natural, as the measurements resulting from different patterns exhibit a random variation. This is due partly to the measurement noise of the measuring devices and partly to the distinct characteristics of each pattern. A typical assumption is that the values associated with each class are distributed according to the Gaussian distribution. When the probability density functions are Gaussian, the n-dimensional Gaussian probability density function has the form given in equation 2.5.

$$p(x/\omega_j) = \frac{1}{(2\pi)^{n/2}\left|\sum_j\right|^{1/2}} e^{-\frac{1}{2}\left[(x-m_j)^T \sum_j^{-1}(x-m_j)\right]}$$

2.5

Where $C_j$ and $m_j$ are the covariance matrix and mean vector of the pattern population of class $\omega_j$ and $\left|\sum_j\right|$ is the determinant of $C_j$

From Bayes Theorem;

$$\text{posterior probability} = \frac{(\text{prior probability})(\text{likelihood probability})}{\text{evidence probability}}$$

Posterior $(\omega_j/\ x) = P\ (\omega_j)\ p(x/\omega_j)/p(x)$.

2.6

The evidence may be ignored since it is a positive constant. In practice we are only interested in the numerator of that fraction, since the denominator does not depend on classes and the values of the features are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model.

We wish to determine which posterior is greatest. For the classification as Fighter aircraft, the first class, the posterior is given by equation 2.7,

$$posterior(Fighter) = \frac{\mathrm{P(Fighter)p(\Phi_1/Fighter)p(\Phi_2/Fighter)} \ldots\ldots \mathrm{p(\Phi_7/Fighter)Prior(Fighter)}}{\mathrm{Evidence}}$$

2.7

For the classification of F35A aircraft.

$$posterior(Bomber) = \frac{\mathrm{P(Bomber)p}(\Phi_1/\mathrm{Bomber})\mathrm{p}(\Phi_2/\mathrm{Bomber}) \ldots\ldots \mathrm{p}(\Phi_7/\mathrm{Bomber})\mathrm{Prior(Bomber)}}{\mathrm{Evidence}} \quad 2.8$$

For the classification of the B2 aircraft. In practice we are only interested in the numerator of the fractions, since the denominator does not depend on the classes but on the given data and the values of the features $\Phi_1$ are given, so that the denominator is effectively positive constant. Where $\Phi_i$ are the invariant moments features.

# CHAPTER 3

# IMAGE PATTERN FEATURES

In this chapter a review of the image pattern feature extraction techniques is discussed. Several qualitative and quantitative techniques have been developed for characterizing the shape of objects within an image. These techniques are useful for classifying objects in a pattern recognition system and for symbolically describing objects in an image understanding system. Shape parameters, one dimensional image functions, chain chord, shock graphs, moments and shape transforms are discussed in relation to feature extraction.

## 3.1. Shape Parameters

Shape image recognition involves a measure of similarity or dissimilarity of feature vectors representing those shapes. Geometric features can be used to discriminate shapes that exhibit large dissimilarities. Therefore they are used as filters or combined with other shape descriptors. They do not have enough discrimination power as shape descriptors. Shapes may be described with many parameters such as the centroid, axis of least inertia, digital bending energy, eccentricity, circularity ratio, elliptic variance, rectangularity, convexity, solidity, Euler number, profiles, holes area ratio. [2] [37] [38] .The following are a few examples of shape description.

### 3.1.1. Centroid Position

The spatial moment concept can be extended to discrete images by forming spatial summations over a discrete image function f(j, k). The (m, n)$^{th}$ spatial geometric moment is defined as;

$$M_U(m,n) = \sum_{j=1}^{J}\sum_{k=1}^{K}\left(x_j\right)^m (y_k)^n f(j,k) \qquad 3.1$$

The (m, n)$^{th}$ scaled spatial geometric moment is then defined as;

$$M(m,n) = \frac{1}{J^m K^n}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(x_j\right)^m (y_k)^n f(j,k) \qquad 3.2$$

$$M(m,n) = \frac{M_U(m,n)}{J^m k^n} \qquad 3.3$$

The ratios

$$x_c = \frac{M(1,0)}{M(0,0)} \qquad 3.4$$

$$y_c = \frac{M(1,0)}{M(0,0)} \qquad 3.5$$

Of first- to zero-order spatial moments define the image centroid. The centroid, is the balance point of the image function such that the mass of
f(j, k) left and right of $x_j$ and above and below $y_k$ is equal [10].

3.1.2. Profiles

If shape image is projected on the Cartesian coordinates system one dimensional functions are obtained. These projections are known as image profiles [37] [39] [40].

$$P_x = \sum_{y=xmin}^{ymax} f(x,y) \quad 3.6$$

$$P_y(y) = \sum_{x=imin}^{xmax} f(x,y) \quad 3.7$$

Where f (x, y) represents the shape region.

## 3.2. Shape Signatures

Shape signature is a 1-D functional representation of a boundary. It is generated in various ways. Complex coordinates, centroid distance function, tangent angle, curvature function, area function, triangle-area representation and chord length function are some of the often used shape signatures. The shape signature can be used as a shape descriptor in preprocessing in other feature extraction algorithms, for example, Fourier descriptors, wavelet description [2] [39]. The following are examples of shape signatures.

### 3.2.1. Complex coordinates

The complex numbers generated from an image boundary points form a complex coordinates function.

$$z(n) = [x(n) - c_x] + j[y(n) - c_y] \quad 3.8$$

where ($c_x$, $c_y$) is the centroid of the shape [1].

### 3.2.2. Centroid distance function

$$f(t) = [x(t) - x_c] + j[y(t) - y_c] \quad 3.9$$

where ($x_c$, $y_c$) is the centroid of the shape, which is the average of the boundary coordinates.

$$x_c = \frac{1}{N}\sum_{t=0}^{N-1} x(t) \quad 3.10$$

$$y_c = \frac{1}{N}\sum_{t=0}^{N-1} y(t) \quad 3.11$$

### 3.2.3. Tangent angle

A contour is described in terms of the arc length between a given point and the origin, within the image. As the length *t* at a given point, the number of consecutive pixels is defined between the given point and the point considered as the origin. The contour description can be achieved via a one-parameter real-valued function, the arc tangent angle θ(t) or the corresponding curvature k(t), defined as;

$$\theta(t) = tan^{-1}\left[\frac{dy\,(t)}{dx\,(t)}\right] \qquad 3.12$$

$$k(t) = \frac{d\theta\,(t)}{dt} \qquad 3.13$$

where x(t), y(t) are the coordinates of the respective point as a function of length t from the origin and;

$$dt = \sqrt{dy^2 + dx^2} \qquad 3.14$$

### 3.2.4. Perimeter and Area

Perimeter and area measurements are meaningful only for binary images. Consider a discrete binary image containing one or more objects, where

*f(j, k)=1* if a pixel is part of the object and *f(j, k)=0* for all non-object or background pixels. The perimeter of each object is the count of the number of pixel sides traversed around the boundary of the object starting at an arbitrary initial boundary pixel and returning to the initial pixel. The area of each object within the image is simply the count of the number of pixels in the object for which *f(j, k)=1.* The enclosed area of an object is defined to be the total number of pixels for which *f(j, k)=0 or 1* within the outer perimeter boundary $P_E$ of the object. The enclosed area can be computed during a boundary-following process while the perimeter is being computed. Assume that the initial pixel in the boundary-following process is the first black pixel encountered in a raster scan of an image. Then, proceeding in a clockwise direction around the boundary, a crack code *C(p),* is generated for each side p of the object perimeter such that *C(p) = 0, 1, 2, 3* for directional angles 0, 90, 180, 270°, respectively. The enclosed area is

$$A_E = \sum_{p=1}^{P_E} j(p-1)\Delta k(p) \qquad 3.15$$

where $P_E$ is the perimeter of the enclosed object and

$$j(p) = \sum_{i=1}^{p} \Delta j\,(i) \qquad 3.16$$

with j(0) = 0. The delta terms are defined by

$$\Delta j(p) = \begin{cases} 1 & If\ C(p) = 1 \\ 0 & If\ C(p) = 0\ or\ 2 \\ -1 & If\ C(p) = 3 \end{cases} \quad 3.17$$

$$\Delta k_p = \begin{cases} 1 & If\ C(p) = 0 \\ 0 & If\ C(p) = 1\ or\ 3 \\ -1 & If\ C(p) = 2 \end{cases} \quad 3.18$$

### 3.2.5. Chord length function

A widely used shape description is contour function. It reduces a *2-D* function into a *1-D* function which is easier to handle than the original shape. There are two types of chord length functions, distance verses angle and curvature functions. Distance verses angle functions uses angle from a given direction as its variable. The curvature function uses the arc length from a starting point as a variable. Curvature function exists for any contour [37].

## 3.3. Chain code

The chain code is a technique of representing a binary object by encoding only the boundary. The chain code is composed of a sequence of numbers between 0 and 7. Each number represents the transition between two consecutive boundary pixels, 0 being a step to the right, 1 a step diagonally right/up, 2 a step up, etc. The following figure contains the directions associated to each code:

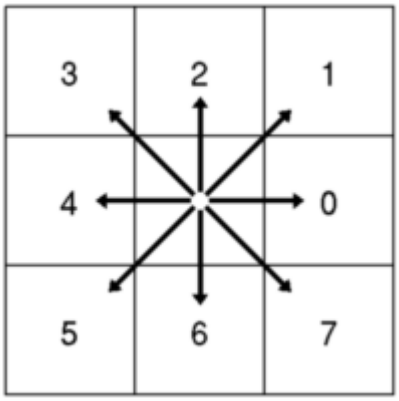



*Fig 3.1 Chain code directions [37].*

The chain code thus has as many elements as there are boundary pixels. It can be used to encode the shape of the object, but not its location. Therefore there is need to remember the coordinates of the first pixel in the chain to solve that [37].

### 3.3.1. Vertex chain code (VCC)

An element of the *VCC* indicates the number of cell vertices, which are in touch with the bounding contour of the shape in that element's position. Only three elements *"1", "2" and "3"* can be used to represent the bounding contour of a shape composed of pixels in the rectangular grid. Figure 3.2 shows the elements of the *VCC* to represent a shape. An element of the *VCC* indicates the number of cell vertices, which in contact with the bounding contour of the shape in that element position. On figure 3.2, starting from the dot the code indicated is *11212113*.

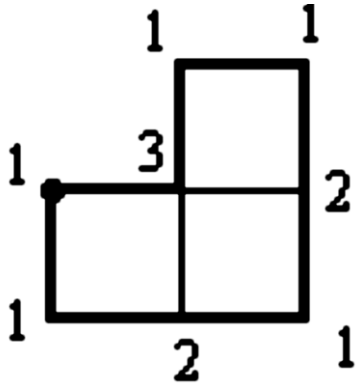



*Fig. 3.2 Vertex chain code [37].*

### 3.3.2. Axis of least inertia

The axis of least inertia of a shape is defined as the line for which the integral of the square of the distances to points on the shape is a minimum. It serves as a unique reference line to preserve the orientation of the shape. The preservation of shape orientation of a shape is essential for extracting features, which are invariant to linear transformations. The slope angle $\theta$ of the axis of least inertia of a shape curve is estimated. Once the slope angle $\theta$ is calculated, each point on the shape curve is projected on to the axis of least inertia. Given a point $(x_i, y_i)$ on a shape curve, the co-ordinates of its projected point ($A_x$ , $A_y$ ) on the axis of least inertia (Fig. 3.3) is located.

The axis minimizes the sum of the squared distance from its position to the boundary of the shape. It is defined by equation 3.19.

$$I(\alpha, S) = \int_S \int r^2(x, y, \alpha)\ dx\, dy \qquad 3.19$$

r (x, y, $\alpha$) is the perpendicular distance from the point (x, y) to the line given by x sin $\alpha$ - y cos $\alpha$ =0. The coordinate (0, 0) is the location of the centroid.

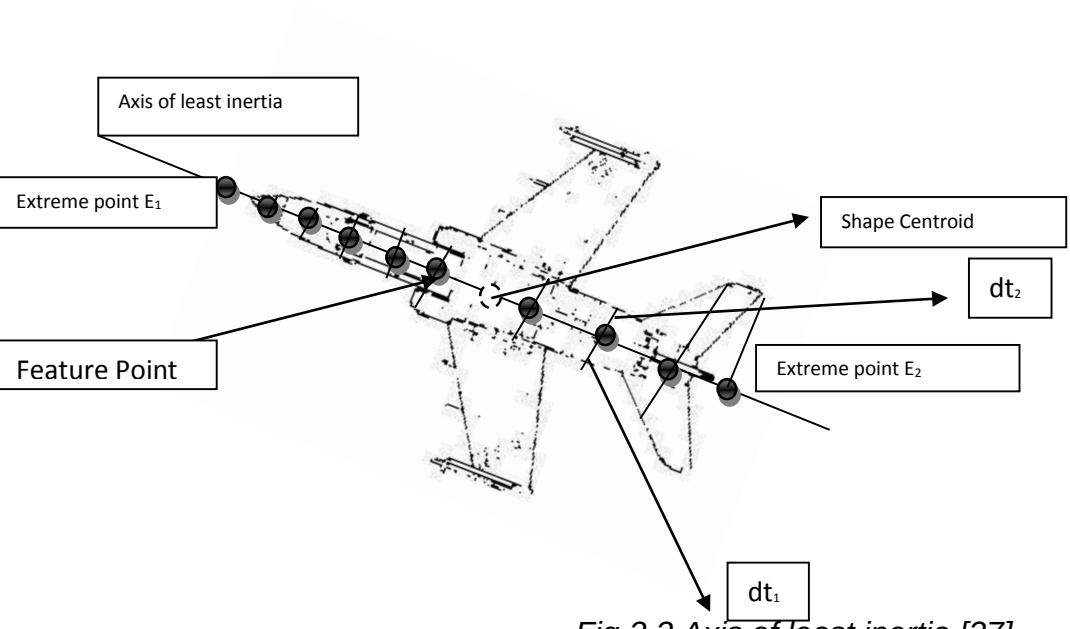



*Fig 3.3 Axis of least inertia [37]*

### 3.4. Moments

Moments of images provide efficient local descriptors and have been used extensively in image analysis applications. Their main advantage is their ability to provide invariant measures of shape.

#### 3.4.1. Boundary moments

The boundary moments can be used to reduce the dimension of boundary representation. If it is a *1-D* shape representation *z(i),* then the $r^{th}$ moment $m_r$ and central moment $\mu_r$ can be estimated as indicated in equation 3.20.

$$m_r = \frac{1}{N}\sum_{i=0}^{n}|z(i)|^r$$

and

$$\mu_r = \frac{1}{N}\sum_{i=1}^{N}|z(i) - m_1|^r$$

3.20

Where *N* is the number of boundary points. The normalized moments are invariant to shape translation, rotation and scaling. Less noise-sensitive shape descriptors can be obtained from normalizing the higher central moment function as given in equation 3.21 [41]

$$F_1 = \frac{(\mu_2)^{1/2}}{m_1} \qquad F_2 = \frac{\mu_3}{{\mu_2}^{3/2}} \qquad \textit{and} \qquad F_3 = \frac{\mu_4}{{\mu_2}^2} \qquad 3.21$$

The other approaches of boundary moments treats the *1-D* shape feature function *z(i)* as a random variable ***v*** and creates a *K* bins histogram $p(v_i)$ from *z(i)*. Then, the $r^{th}$ central moment is obtained [27] as given in equation 3.22

$$\mu_r = \sum_{i=1}^{K} (v_i - m)^r p(v_i)$$

*and*

$$m = \sum_{i=1}^{K} v_i p(v_i)$$

3.22

The advantage of boundary moment descriptors is that they are easy to implement. However, it is difficult to associate higher order moments with some physical interpretation.

### 3.4.2. Region moments

Among the region-based descriptors, moments are utilized extensively. These include invariant moments, Zernike moments, Radial Tchebyshev moments, Legendre moments, Polynomial moments, Poisson Fourier-Mellin etc. [42]

The general form of a moment function $m_{pq}$ of order ($p$ + $q$) of a shape region can be given as:

$$m_{pq} = \sum_{x} \sum_{y} \psi_{pq}(x, y) f(x, y)$$

*p,q = 0,1,2,3……..* 3.23

where $\Psi_{pq}$ as in the equation 3.23 is known as the moment weighting kernel or the basis set; The function *f(x, y)* is the shape region.

#### 3.4.2.1. Invariant moments

Invariant moments (IM) are also called geometric moment invariants and have a basis $\Psi_{pq} = x^p y^q$. They are not orthogonal. The geometric moment function $m_{pq}$ of order ($p$ + $q$) [43] is given as:

$$m_{pq} = \sum_{x} \sum_{y} x^p y^q f(x, y)$$

*p,q = 0,1,2……* 3.24

The geometric central moments, which are invariant to translation, are defined as [42]

$$\mu_{pq} = \sum_{x} \sum_{y} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

*p,q= 0,1,2……...* 3.25

Where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$ 3.26

A set of 7 invariant moments have been derived and shown to offer complete invariance as given in equation 3.27 [1] [2] [3] [4] [5].

$$\emptyset_1 = \eta_{20} + \eta_{02}$$

$$\emptyset_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\emptyset_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\emptyset_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\emptyset_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}).[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\emptyset_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}^2(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\emptyset_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03}).[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Where $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$ and $\gamma = 1 + \frac{(p+q)}{2}$ *for p+q=2,3……...* 3.27

Invariant moments are computationally simple and are also invariant to rotation, scaling and translation. However, they have several drawbacks:

(i) Information redundancy due to non orthogonality of the basis.

(ii) Higher-order moments are very sensitive to noise.

(iii) Large variation in the dynamic range of values: since the basis involves powers of *p* and *q*, the moments computed have large variation in the dynamic range of values for different orders. This may cause numerical instability when the image size is large.

#### 3.4.2.2. Algebraic moment invariants

The algebraic moment invariants are computed from the first *m* central moments and are given as the eigenvalues of predefined matrices, $M_{[j,k]}$, whose elements are scaled factors of the central moments. They are not orthogonal. The algebraic moment invariants can be constructed up to arbitrary order and are invariant to affine transformations. However, algebraic moment invariants performed either very well or very poorly on the objects with different configuration of outlines. [2]

#### 3.4.2.3. Zernike moments

Zernike moments are a class of orthogonal moments and have been shown to be effective in terms of image representation. Zernike moments, a type of moment function, are the mapping of an image onto a set of complex Zernike polynomials. As these Zernike polynomials are orthogonal to each other, Zernike moments can represent the properties of an image with no redundancy or overlap of information between the moments. The Zernike complex moments are as given in equation 3.30 [8] using equation 3.28 and 3.29.

$$V_{nm}(x,y) = V_{nm}(rcos\theta, sin\theta) = R_{nm}(r)exp(jm\theta) \qquad 3.28$$

where $R_{nm}(r)$ is the orthogonal radial polynomial.

$$R_{nm}(r) = \sum_{s=0}^{\frac{(n-|m|)}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n-2s+|m|}{2}\right)!\left(\frac{n-2s-|m|}{2}\right)!} r^{n-2s} \qquad 3.29$$

$n = 0, 1, 2, \ldots\ 0 \leq |m| \leq n$; and $n - |m|$ is even. The Zernike polynomials are a complete set of complex valued functions orthogonal over the unit disk, i.e $x^2 + y^2 \leq 1$. The Zernike moment of order $n$ with repetition $m$ of shape region described by the function $f(x, y)$ is given by equation 3.30 [30].

$$Z_{nm} = \frac{n+1}{\pi} \sum\sum f(rcos\theta, rsin\theta).R_{nm}(r).exp(jm\theta) \quad r \leq 1 \qquad 3.30$$

Expressed in a polar coordinate system.

The Zernike moments have the following advantages:

(i) Rotation invariance: The magnitudes of Zernike moments are invariant to rotation.

(ii) Robustness: They are robust to noise and minor variations in shape.

(iii) Expressiveness: Since the basis is orthogonal, they have minimum information redundancy.

However, the computation of Zernike moments pose several problems as do many of the continuous orthogonal moments.

(i) The image coordinate space must be transformed to the unit circle domain where the orthogonal polynomial is defined.

(ii) The continuous integrals must be approximated by discrete summations. This approximation not only leads to numerical errors in the computed moments, but also severely affects the analytical properties such as rotational invariance and orthogonality.

(iii) The computational complexity of the radial Zernike polynomial increases as the order becomes large.

### 3.4.6. Radial Tchebyshev moments

The radial Tchebyshev moment of order *p* and repetition *q* is defined as [11]:

$$S_{pq} = \frac{1}{2\pi\rho(p,m)} \sum_{r=0}^{m-1} \sum_{\theta=0}^{2\pi} t_p(r).exp(-jq\theta).f(r,\theta)$$

3.31

where $t_p(r)$ is the scaled orthogonal Chebyshev polynomials for an image of size $N \times N$ and where;

$$t_0(x) = 1$$

$$t_1(x) = \frac{(2x - N + 1)}{N}$$

$$t_p(x) = \frac{(2p-1)t_1(x)t_{p-1}(x) - (p-1)\left\{1 - \frac{(p-1)^2}{N^2}\right\}}{p} t_{p-2}(x)$$

$for$ p>1 3.32

$\rho(p,N)$ is the squared-norm:

$$\rho(p,N) = \frac{N\left(1 - \frac{1}{N^2}\right)\left(1 - \frac{2^2}{N^2}\right)...\left(1 - \frac{p^2}{N^2}\right)}{2p+1}$$

*p=0,1,…,N-1* 3.33

*and m = (N/2) + 1.*

The mapping between ($r$, $\theta$) and image coordinates ($x$, $y$) is given by:

$$x = \frac{rN}{2(m-1)}\cos\theta + \frac{N}{2}$$

$$y = \frac{rN}{2(m-1)}sin\theta + \frac{N}{2}$$

3.34

Radial Chebyshev moments possess rotational invariance property.

### 3.4.7. Legendre Moments

Orthogonal Legendre moments are used in pattern recognition and image processing applications. Translation and scale Legendre moment invariants are achieved directly by using Legendre polynomials. The Legendre moments are continuous orthogonal moments. They can be used to attain a near zero value of redundancy measure in a set of moment functions, so that the moments correspond to independent characteristics of the image. The translation and scale invariant functions of Legendre moments are achieved by using a

combination of the corresponding invariants of geometric moments. They can also be accomplished by normalizing the translated and/or scaled images using complex or geometric moments. However, the derivation of these functions is not based on Legendre polynomials. This is mainly due to the fact that it is difficult to extract a common displacement or scale factor from Legendre polynomials. The two dimensional Legendre moments of order (p + q), with image intensity function f(x, y), are defined as equation 3.35: [8]

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^{1} \int_{-1}^{1} P_p(x) P_q(y) f(x,y) dx dy$$

$$where \;\; x, y \in [-1,1]$$

3.35

Where Legendre polynomial, $P_p(x)$, of order p is given by equation 3.36:

$$P_p(x) = \sum_{k=0}^{p} (-1)^{\frac{p-k}{2}} \frac{1}{2^p} \frac{(p+k)!\, x^k}{\left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)! \, k!}$$

3.36

The recurrence relation of Legendre polynomials, $P_p(x)$, is given as follows:

$$P_p(x) = \frac{(2p-1) x P_{p^{-1}}(x) - (p-1) P_{p^{-2}}(x)}{p}$$

3.37

Where $P_o(x) = 1$, $P_1(x) = x$ and $p>1$. Since the region of definition of Legendre polynomials is the interior of [-1, 1], a square image of N x N pixels with intensity function f(i, j), $0 \leq i, j \leq (N - 1)$, is scaled in the region of $-1 < x, y < 1$. In the result of this, equation (3.37) can now be expressed in discrete form as: [8]

$$L_{pq} = \lambda_{pq} \sum_{i-0}^{N-1} \sum_{j=0}^{N-1} P_p(x_i) P_q(y_j) f(i,j)$$

3.38

where the normalizing constant,

$$\lambda_{pq} = \frac{(2p+1)(2q+1)}{N^2}$$

3.39

$x_i$ and $y_i$ denote the normalized pixel coordinates in the range of [-1, 1], which are given by:

$x_i = \frac{2i}{N-1} - 1$ And $y_j = \frac{2j}{N-1} - 1$

3.40

## 3.5. Shape transform domains

These are methods which are formed by the transform of the detected object or of the whole image. Transforms can then be used to characterize the appearance of images. The shape feature is represented by the all or partial coefficients of a transform.

### 3.5.1. Fourier descriptors

The perimeter of an arbitrary closed curve can be represented by its instantaneous curvature at each perimeter point. A point on the perimeter is measured by its polar position *z(s)* as a function of arc length *s.* The complex function *z(s)* may be expressed in terms of its real part *x(s)* and imaginary part *y(s)* as expressed in equation 3.41 [44]

$$z(s) = x(s) + jy(s)$$

3.41

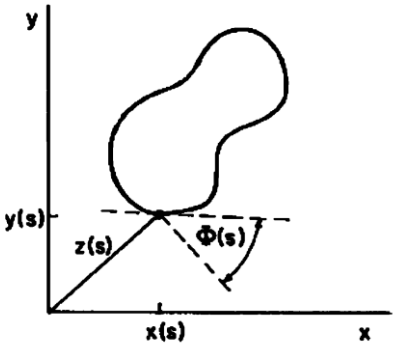



*Fig 3.4 Tangent angle [37]*

The tangent angle defined in Figure 3.4 is given by

$$\varphi(s) = arctan\left\{\frac{dy(s)/dy}{dx(s)/dx}\right\}$$

3.42

and the curvature is the real function given by equation 3.43

$$k(s) = \frac{\varphi(s)}{ds}$$

3.43

The coordinate points [*x*(*s*), *y*(*s*)] can be obtained from the curvature function by the reconstruction formulas 3.44

$$x(s) = x(0)\int_0^s k\,(\alpha)\cos\{\varphi(\alpha)\}d\alpha$$

$$y(s) = y(0)\int_0^s k(\alpha)sin\{\varphi(\alpha)\}\,d\alpha$$

3.44

Where *x*(0) and *y*(0) are the starting point coordinates.

#### 3.5.1.1 One-dimensional Fourier descriptors

In general, a Fourier descriptor is obtained by applying the Fourier transform on a shape signature derived from shape boundary coordinates. The normalized Fourier transformed coefficients are termed the Fourier descriptor of the shape. Fourier descriptors derived from different signatures have significant different performance on shape retrieval. For example, Fourier descriptor derived from centroid distance function *r(t)* outperforms Fourier descriptor derived from other shape signatures in terms of overall performance. [37] The discrete Fourier transform of *r(t)* is then given by equation 3.45:

$$a_n = \frac{1}{N}\sum_{t=0}^{N-1} r(t)exp\left(\frac{-j2\pi nt}{N}\right)$$

n=0,1,……,N-1 3.45

Since the centroid distance function *r(t)* is only invariant to rotation and translation, the acquired Fourier coefficients have to be further normalized so that they are scaling and start point independent shape descriptors. From Fourier transform theory, the general form of the Fourier coefficients of a contour centroid distance function *r*(*t*) transformed through scaling and change of start point from the original function $r(t)^{(o)}$ is given by equation 3.46: [12][45]

$$b_n = \frac{a_n}{a_1} = \frac{exp(jn\tau).s.{a_n}^0}{exp(j\tau).s.{a_1}^0}$$

$$= \frac{{a_n}^0}{{a_1}^0}exp[j(n-1)\tau]$$

$$= {b_n}^0 exp[j(n-1)\tau]$$

$Where\ \tau$ is the angle incurred by the change of start point.

3.46

where $b_n$ and $b_n^{(o)}$ are the normalized Fourier coefficients of the transformed shape and the original shape, respectively. If we ignore the phase information and only use magnitude of

the coefficients, then $|b_n|$ and $|b_n^{(o)}|$ are the same. In other words, $|b_n|$ is invariant to translation, rotation, scaling and change of start point.

The set of magnitudes of the normalized Fourier coefficients of the shape $\{|b_n|, 0 < n < N\}$ are used as shape descriptors, denoted as:

$$\{b_n, 0 < n < N\}$$

One-dimensional Fourier descriptor has several nice characteristics such as simple derivation, simple normalization and simple to do matching. For efficient retrieval, several Fourier descriptors are sufficient for shape description.

### 3.5.2. Region-based Fourier descriptor

The region-based Fourier descriptor is referred to as generic Fourier descriptor, which can be used for general applications. Basically, generic Fourier descriptor is derived by applying a modified polar Fourier transform on shape image [33],[34]. In order to apply modified polar Fourier transform, the polar shape image is treated as a normal rectangular image. The steps are

(i) The approximated normalized image is rotated counter clockwise by an angular step sufficiently small.

(ii) The pixel values along positive x-direction starting from the image center are copied and pasted into a new matrix as row elements.

(iii) The steps 1 and 2 are repeated until the image is rotated by 360°.

The result of these steps is that an image in polar space plots into Cartesian space.The transformation of a polar image into a Cartesian coordinate image is shown in figure 3.5.

*Fig 3.5 Polar into Cartesian coordinate image. [45]*

The Fourier transform is acquired by applying a discrete 2D Fourier transform on this shape image:

$$p_f(\rho, \theta) = \sum_r \sum_i f(r, \theta_i) exp\left[j2\pi\left(\frac{r}{R}\rho + \frac{2\pi i}{T}\phi\right)\right] \quad 3.47$$

Where

$$0 \le r = \sqrt{\left[(x - g_x)^2 + \left(y - g_y\right)^2\right]} < R$$

and

$$\theta_i = i\left(\frac{2\pi}{T}; 0 \le \rho < R, 0 \le \emptyset < T.\left(g_x, g_y\right)\right) \quad 3.48$$

is the centroid of the shape; *R* and *T* are the radial and angular resolutions. The acquired Fourier coefficients are translation invariant. Rotation and scaling invariance are achieved by the following generic Fourier descriptor equation, which is given by equation 3.49

$$f_g = \left\{\frac{\left|p_f(0,0)\right|}{area}, \frac{\left|p_f(0,1)\right|}{\left|p_f(0,0)\right|}, \ldots \frac{\left|p_f(0,n)\right|}{\left|p_f(0,0)\right|}, \ldots, \frac{\left|p_f(m,0)\right|}{\left|p_f(0,0)\right|}, \ldots \frac{\left|p_f(m,n)\right|}{\left|p_f(0,0)\right|}\right\} \quad 3.49$$

Where *area* is the area of the bounding circle in which the polar image resides. The maximum number of the radial frequencies selected is given as *m* and *n* is the maximum number of angular frequencies selected. To achieve hierarchical coarse to fine representation requirement *m* and *n* can be adjusted. The experimental results have shown Generic Fourier descriptor as invariant to translation, rotation, and scaling [41] [44] [45].

# CHAPTER 4

# BAYESIAN DECISION BASED IMAGE RECOGNITION

The Bayesian classification method is one of the most popular probabilistic based techniques and will be presented here. The approach to be followed builds upon probabilistic arguments stemming from the statistical nature of the generated features. As has already been pointed out in the introductory chapter, this is due to the statistical variation of the patterns as well as to the noise in the measuring sensors. The classification of an unknown pattern into the most probable class will be performed. Thus, the task now becomes that of defining what “most probable” means.

## 4.1. Bayes Classifier

Bayes classifier is a statistical technique based on Bayes' theorem which is given by the relationship in equation 4.1.

$$\mathrm{P}(A_j/B) = \frac{\mathrm{P}(B/A_j)\mathrm{P}(A_j)}{P(B)} \qquad 4.1$$

$P(A_j)$: Is the prior belief (probability of hypothesis before seeing any data).

$P(B/A_j)$: Is the likelihood (probability of the data if the hypothesis is true.

The probability of the data through observation is given by equation 4.2.

$$P(B) = \sum_j P(B/A_j)\, P(A_j) \qquad 4.2$$

$P(A_j/B)_j$: Is the posterior (probability of hypothesis having seen the data)

Bayes classifier has decision density functions of the form.

$$d_j(x) = p\left({}^{x}/_{\omega_j}\right) P(\omega_j)$$

$$j = 1,2,\ldots\ldots\ldots\ldots W \qquad 4.3$$

where $p\left({}^{x}/_{\omega_j}\right)$ is the probability density function (pdf) of the pattern vectors of class $\omega_j$, and $P(\omega_j)$ is the probability that class $\omega_j$ occurs. Given an unknown pattern vector, the process

is to compute a total of *W* decision functions and then assign the pattern to the class whose decision function yielded the largest numerical value. The case where the pdf is Gaussian is of particular interest. The *n*-dimensional Gaussian pdf has the form given in equation 4.4

$$P(x/\omega_j) = \frac{1}{(2\pi)^{n/2}|\sum|^{1/2}} e^{-\frac{1}{2}\left[(x-m_j)^T \sum^{-1}(x-m_j)\right]}$$

4.4

Where ∑and $m_j$ are the covariance matrix and mean vector of the pattern population of class $\omega_j$ and |∑| is the determinant of ∑. [46]

Given a classification task of M classes, $\omega_1, \omega_2, \omega_3, \ldots \omega_m$ and an unknown pattern, which is represented by a feature vector *x,* then *M* conditional probabilities $p\left({}^{\omega_j}/_{x}\right)$, j = 1, 2, 3……M are formed. These are also referred to as a posteriori probabilities. Each of them represents the probability that the unknown pattern belongs to the respective class $\omega_j$, given that the corresponding feature vector takes the value *x*. Indeed, the Bayes' classifier computes either the maximum of these M values or, equivalently, the maximum of an appropriately defined function of them. The unknown pattern is then assigned to the class corresponding to this maximum [41].

### 4.2. Approaches to statistical pattern recognition

In pattern classification, the goal is to assign a pattern vector *x* that represent a set of measurements to one of *M* possible classes $\omega_i$ , where

i = 1,…M. A decision rule partitions the measurement space into M regions $\Omega_i$ i = 1,…C. If an observation vector is in $\Omega_i$ then it is assumed to belong to class $\omega_i$.

$$P(\omega_i/(x)) = \frac{P(x/\omega_i)P(\omega_i)}{P(x)}$$

4.5

where *p*(***x***) is the pdf of ***x*** and for which we have;

$$P(x) = \sum_{i=1}^{C} P(x/\omega_i)P(\omega_i)$$

4.6

The Bayes classification rule is stated as;

If P($\omega_1$/x)>P($\omega_2$/x) x is classified to $\omega_1$

P($\omega_2$/x)>P($\omega_1$/x) x is classified to $\omega_2$

Each region may be multiply connected – that is, it may be made up of several disjoint regions. The boundaries between the regions $\Omega_i$ are the decision boundaries or decision surfaces. Generally, it is in regions close to these boundaries that the highest proportion of misclassifications occurs. In such situations, we may reject the pattern or withhold a decision until further information is available so that a classification may be made later. This option is known as the reject option and therefore we have *C + 1* outcomes of a decision rule (the reject option being denoted by $\omega_o$) in a *C*-class problem. There are two approaches to discrimination that will be explored. The first assumes knowledge of the underlying class-conditional probability density functions (the probability density function of the feature vectors for a given class) [21].

### 4.2.1. Supervised versus unsupervised

The two main divisions of classification are supervised classification also referred to as discrimination and unsupervised classification which is simply referred to as clustering. In supervised classification we have a set of data samples (each consisting of measurements on a set of variables) with associated labels, the class types. These are used as exemplars in the classifier design. In unsupervised classification, the data are not labelled and we seek to find groups in the data and the features that distinguish one group from another. A clustering scheme may be applied to the data for each class separately and then representative samples for each group within the class (the group means, for example) used

as the prototypes for that class. It is usual to use the group mean vector as the prototype [21] [22].

### 4.3. Bayes' decision rule for minimum error

Consider *M* classes, $\omega_1, \omega_2, \omega_3, \ldots \omega_M$, with a priori probabilities and we wish to minimize the probability of making an error and there is no information regarding an object other than the class probability distribution then we would assign an object to class $\omega_j$ if;

$$P(\omega_j) > P(\omega_k) \quad k=1,\ldots,M;\ k \neq j \qquad 4.7$$

This classifies all given objects. For classes with equal probabilities, patterns are assigned arbitrarily between those classes. However, we do have an observation vector or measurement vector *x* and we wish to assign *x* to one of the *M* classes. A decision rule based on probabilities is to assign *x* to class $\omega_j$ if the probability of class $\omega_j$ given the observation *x*, $p(\omega_j|x)$, is greatest over all classes $\omega_1, \ldots \omega_M$.

That is, assign x to class $\omega_j$ if;

$$P(\omega_j|x) > P(\omega_k|x) \quad k=1,\ldots,M;\ k \neq j \qquad 4.8$$

This decision rule partitions the measurement space into M regions $\Omega_1, \Omega_2 \ldots \Omega_C$ such that if $x \in \Omega_j$ then *x* belongs to class $\omega_j$ . The a posteriori probabilities $P(\omega_j|x)$ may be expressed in terms of the a priori probabilities and the class-conditional density functions $p(x|\omega_j)$ using Bayes' theorem as:

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)}$$

4.9

and so the decision rule (4.8) may be written as

assign $x$ to $\omega_j$ if;

$$\mathrm{P}(\mathrm{x}|\omega_\mathrm{j})\mathrm{p}(\omega_\mathrm{j}) > P(\mathrm{x}|\omega_\mathrm{k})\mathrm{P}(\omega_\mathrm{k})$$

$k= 1,….,M; k\neq j$

4.10

This is known as Bayes' rule for minimum error. [47]

For two classes, the decision rule (4.10) may be written:

$$l_r(x) = \frac{P(x|\omega_1)}{P(x|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)}$$

implies $x \in$ class $\omega_1$ 4.11

The function $l_r(x)$ is the likelihood ratio. Figures 4.1 and 4.2 give a simple illustration for a two-class discrimination problem. Class $\omega_1$ is normally distributed with zero mean and unit variance,

$$P(x|\omega_1) = N(x|0,1)$$

4.12

Class $\omega_2$ is a normal mixture (a weighted sum of normal densities).

For example.

$$P(x|\omega_2) = 0.6N(x|1,1) + 0.4N(x|-1,2)$$

4.13

Figure 4.1 illustrates plots of $\mathrm{p}(\mathrm{x}|\omega_\mathrm{i})\mathrm{p}(\omega_\mathrm{i})$ where i =1, 2, and the priors are taken to be

$$P(\omega_1) = 0.5\ ,P(\omega_2) = 0.5 \qquad 4.14$$

$$P(x|\omega_i)P(\omega_i)$$

i=1,2, the priors are taken as 0.5

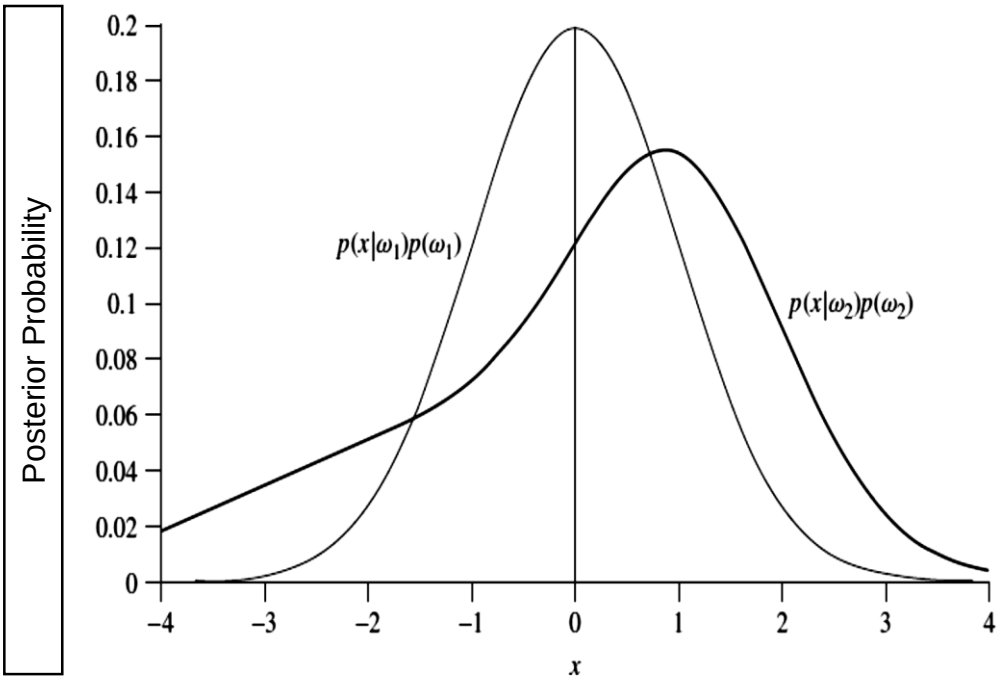



*Fig 4.1 Posterior Probabilities [5]*

Figure 4.2 illustrates plots of the likelihood ratio $\mathrm{L_r(x)}$ and the threshold $P(\omega_2)/P(\omega_1)$. From figure 4.2 the decision rule (4.4) leads to a disjoint region for class $\omega_2$.

The fact that the decision rule (4.2) minimizes the error may be seen as follows. The probability of making an error, *p(*error), may be expressed as:

$$p(error) = \sum_{i=1}^{C} p(error|\omega_i)\, p(\omega_i)$$

4.15

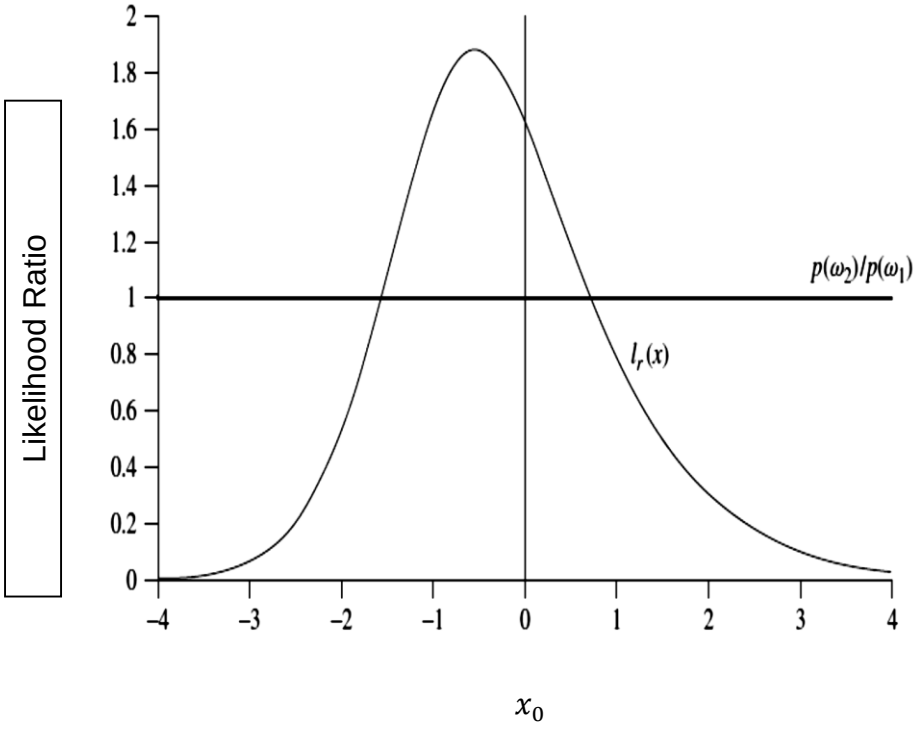



*Fig 4.2 Likelihood ratio $l_r(x)$ and the threshold.*

**4.4. Vector Quantization** Clustering is the task of assigning a set of objects into groups so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other groups. Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Cluster analysis itself is the general task to be solved. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem.

Let *X* be the Data set;

$$X = \{x_1, x_2, x_3 \ldots x_N\} \qquad 4.16$$

*m*-clustering of *X* is defined as the partition of *X* into *m* sets (clusters), $C_1 \ldots\ldots C_m$ so that the following three conditions are met:

i. $C_i \neq 0, \qquad i = 1, \ldots, m$

ii. $\cup_{i=1}^{m} C_i = X$

iii. $C_i \cap C_j = 0, i \neq j,\ i,j = 1,2,3 \ldots\ldots m$

4.17

Typical cluster models include:

i. Connectivity models: for example hierarchical clustering builds models based on distance connectivity.
ii. Centroid models: for example the *k*-means algorithm represents each cluster by a single mean vector.
iii. Distribution models: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the expectation-maximization algorithm.

Vector Quantization is a connectivity model based on the distance between data. However, the goal of VQ is to place points in space in a way that is representative of the data distribution. A Vector Quantization classifier is based on the definition of certain key points called codebook vectors in the data space. Once these relevant points are singled out, the new data is classified to belong to the same class of the closest codebook vector in the Euclidean metric.

For example, consider Figure 4.3, where two codebook vectors are represented with a diamond and with a crossed square. The definition of a codebook vector includes the location in the relevant space and the class that codebook vector belongs to, which can be read in the legend. Accordingly, vector *x* is classified to belong to class 2, while vector *y* belongs to class 1. Basically, the salient features of the data are quantized (i.e. approximated) and represented by the exclusive means of the codebook vectors, which can be thought of as the most representative points of each class. Accordingly, the codebook points define a partitioning of the data space through the midplanes between neighbouring pairs of vectors. For the data represented in Figure 4.3, a good placement of the representative points may look like in Figure 4.4 where the midplanes between neighbouring pairs of vectors would

approximate the optimal Bayesian separation curve. This highlights the key difference between a Bayesian and a VQ classifier: while the Bayesian strategy seeks to approximate the discriminant functions over the whole space used to represent the data, the VQ classifier focuses on a small region of this space, where most of the action takes place. The identification of good codebook vectors may seem difficult without prior knowledge of (or assumptions about) the statistical distributions involved [46] [47] [48] [49].

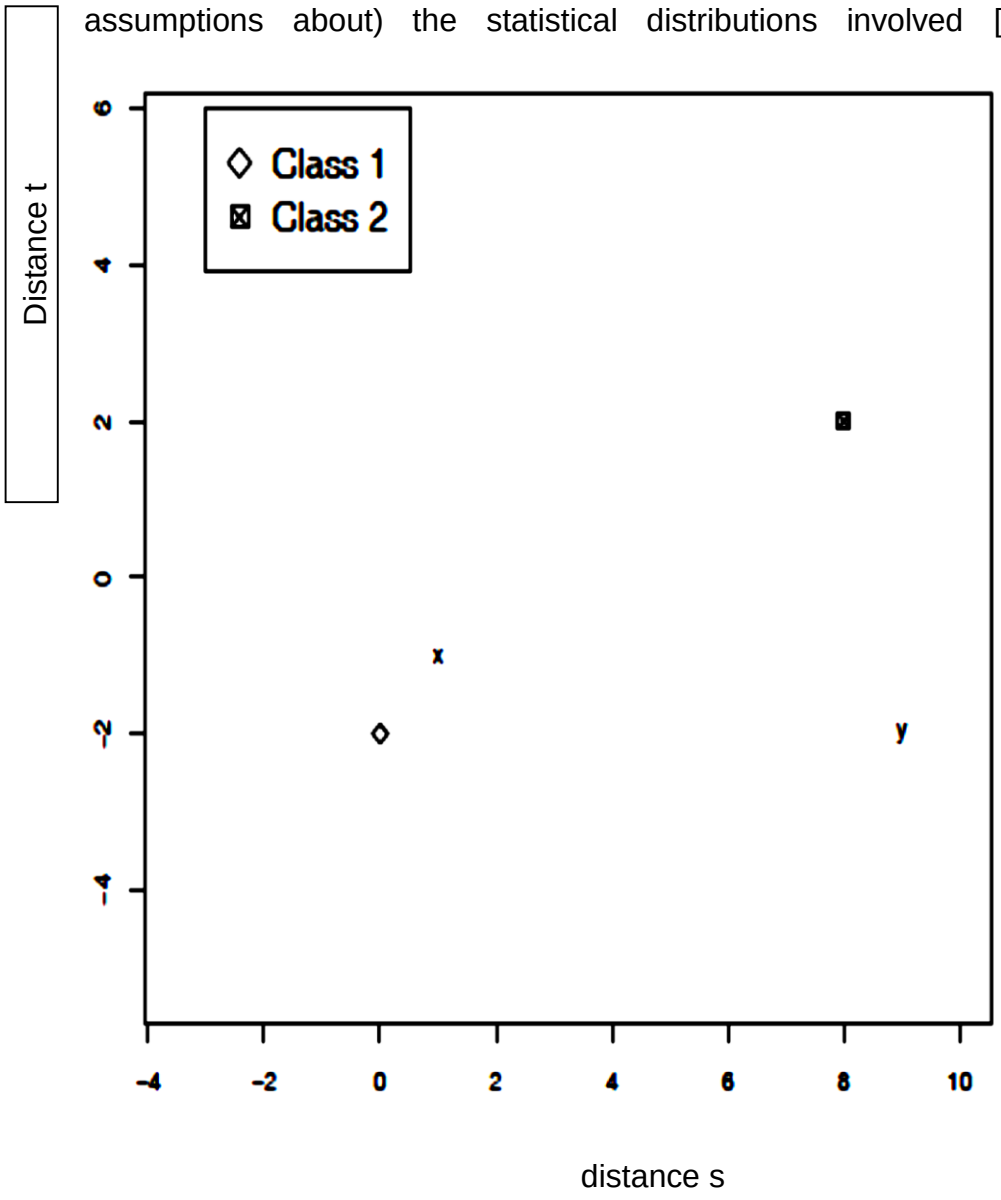



*Fig 4.3 Nearest neighbour classifier rule.*

*Vector x belongs to class 2, while vector y to class 1* [23].

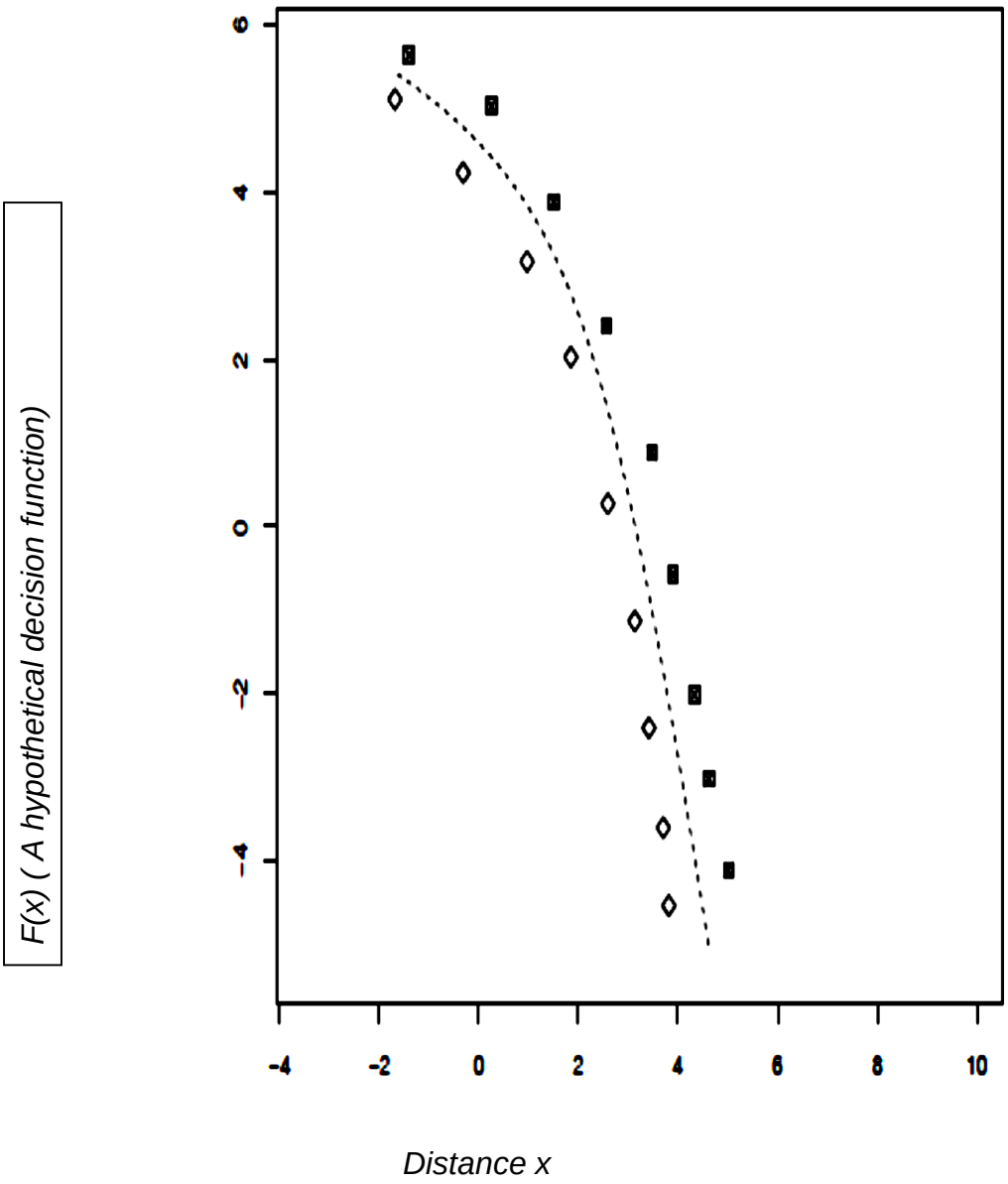



*Fig 4.4 Hypothetical placement of the codebook vectors*

The vectors are for the data shown in Figure 4.3 of certain key points called codebook vectors in the data space [23].

The representative codeword is determined to be the closest in Euclidean distance from the input vector.

Figure 4.5 illustrates codewords in 2-dimensional space. Input vectors are marked with an x, codewords are marked with circles, and the Voronoi regions are separated with boundary lines [50] [51] [52] [53] [54] [55] [56].

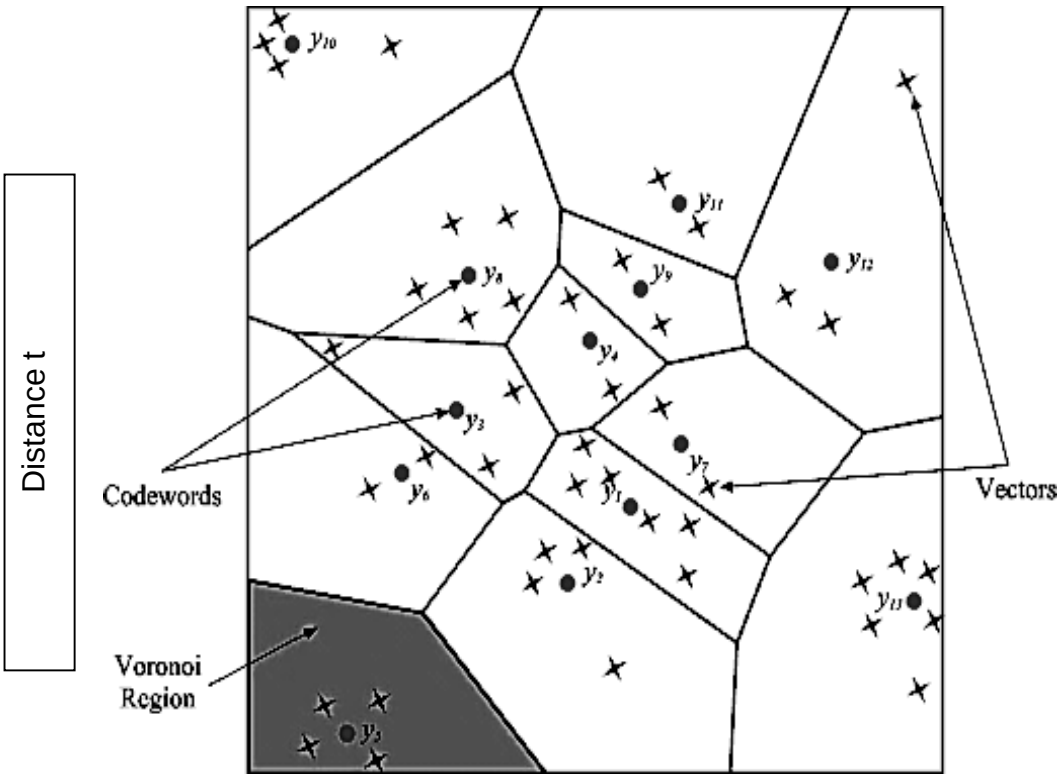



*Fig 4.5 An illustration of Voronoi regions* [23]

Where s and t are Cartesian coordinates.

# CHAPTER 5

# RESULTS AND DISCUSSIONS

The database that has been used in simulating and testing the various techniques is made up of 108 satellite aircraft images taken from both the Federation of American Scientists, and World Military Aircrafts Top View databases. These are standard databases used by researchers and can be accessed through the internet. The former database is made up of all American military aircrafts such as bombers, attack aircrafts, fighter aircrafts, cargo aircrafts, tanker aircrafts, rotary aircrafts, trainer aircrafts, and X-planes. The latter database is made up of the rest of the world military aircrafts. The Federation of American Scientists (FAS) is an organization with the stated intent of using science and scientific analysis to attempt to make the world more secure. The FAS was founded in 1945 by scientists who developed the first atomic bombs who later realized that this precedent had created an extremely dangerous world order. It promotes a safer and more secure world by developing and advancing solutions to important science and technology security problems by educating the public and policy makers.

The images employed were of various sizes ranging from 100X100 to 1024 X 1024 pixels with 8-bits resolution. The simulation of the pattern recognition scheme and the assessment of the performance used the MATLAB version 7.8.0.347 (R2009a) platform.

**5.1 Preprocessing** The reduction of noise is one of the most important processes that can be used to improve the image recognition rate. In satellite images, the degradation in quality is due to speckle noise. Thus, the reduction of speckle noise is often the first stage in a satellite image recognition system. There is no comprehensive method that takes all the constraints in a noisy image into consideration. Filtering is one of the common methods which is used to reduce the speckle noise. The Lee filter has been reported to give favourable results by several researchers. In this thesis it has been recommended based on the statistical and experimental results. Figure 5.1 illustrates two aircraft satellite images before denoising and figure 5.2 illustrates images after pre-processing and binarization.

*Fig 5.1 Unprocessed images.*

### 5.1.1 Denoising Results

Figure 5.2 is a representation of preprocessed that is noise removal and binarization images. The denoising process is based on boundary signature. The programme code is given in Appendix A1.

*Fig 5.2 Typical pre-processed aircraft satellite images*

Satellite images are mostly contaminated by speckle noise. This is multiplicative that can be modelled as: J = I+n*I, where *n* is uniformly distributed random noise with zero mean and variance $\delta^2$. I is the noise free image and J is the noisy image. The effects of filtering with a Lee adaptive filter were simulated using different levels of speckle noise as shown in figure 5.3 then filtered. Their minimum square error (*MSE)* and PSNR (Peak Signal to Noise Ratio) are evaluated as illustrated in table 5.1

*Original Image*

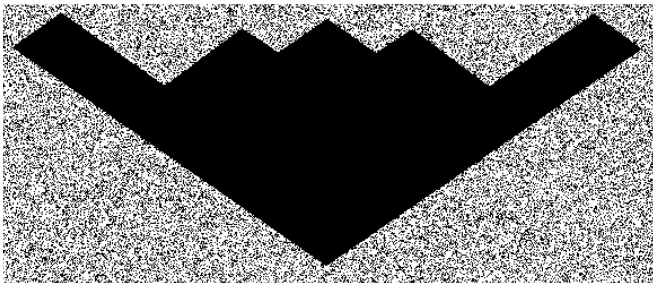


*Image with a 10% noise addition*

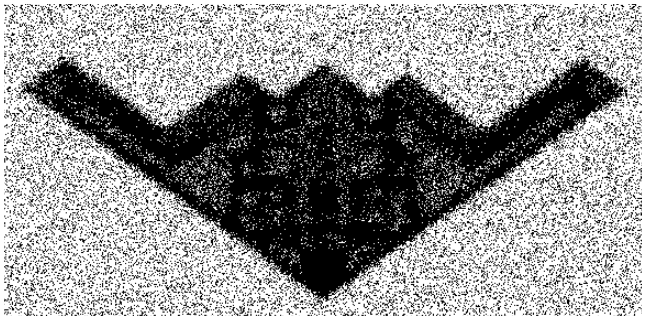


*Image with a 20% noise addition*

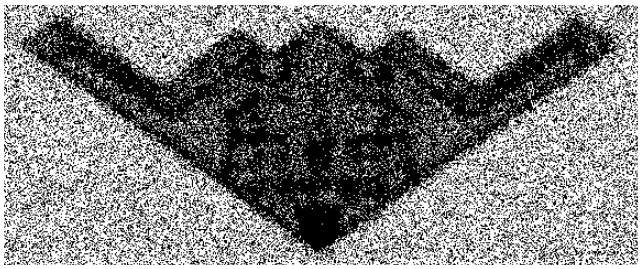


*Image with a 30% noise addition*

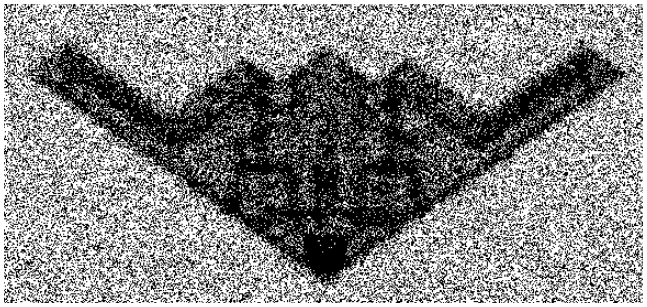


*Image with a 40% noise addition*

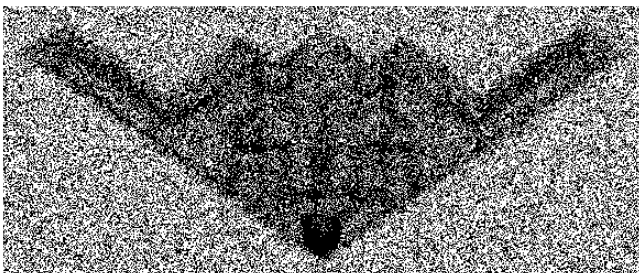


*Image with a 50% noise addition*

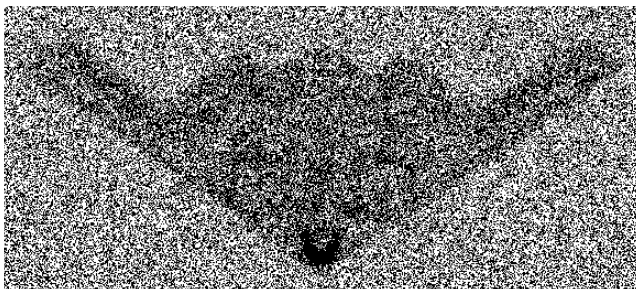


*Image with a 60% noise addition*

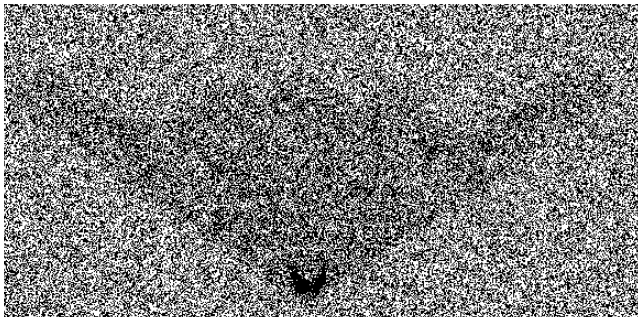


*Image with a 70% noise addition*

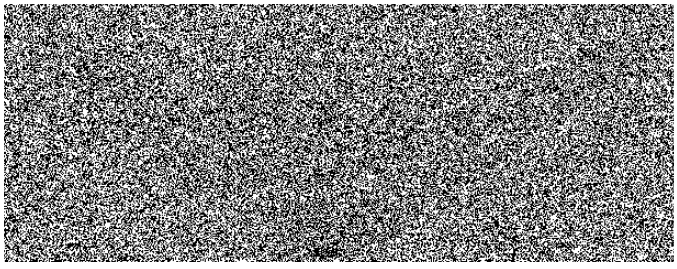


*Image with an 80% noise addition*

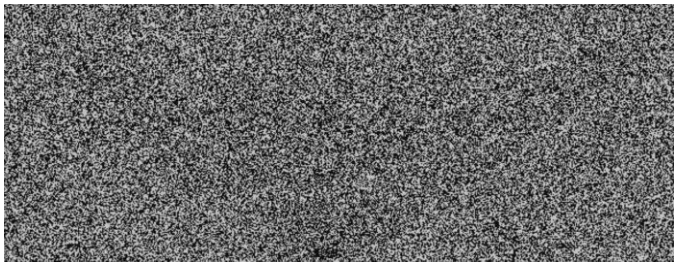


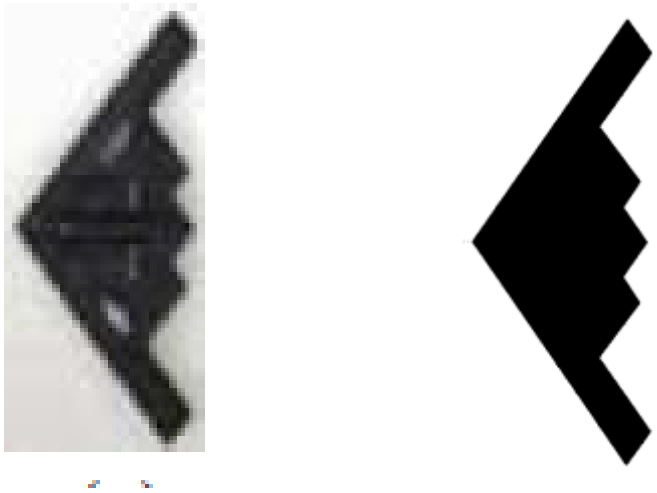
*Image with a 90% noise addition*

*Fig 5.3 Images of B2 bomber with different noise levels*

*Table 5.1 PSNR levels*

| Noise Level Variances | 0.01 | 0.02 | 0.03 | 0.04 | 0.06 | 0.08 |
|---|---|---|---|---|---|---|
| PSNR dB | 25.66 | 22.68 | 20.91 | 19.66 | 18.72 | 17.92 |

The image used for simulation is a standard 512X512 pixel 8 bit depth image with sharp edges as shown in figure 5.4(a). A uniformly distributed multiplicative noise with mean zero and various variances as illustrated is added to the simulated imagery. To test the efficiency of the filter mentioned above, a 5x5 kernel is used for the filter. Previous results have shown that a small kernel of 3X3, 5X5 or 7X7 offer the best results. The filter is applied to the noise contaminated imagery. Figure 5.4(b) shows the filtered image.



*(a) Noisy image* *(b) Denoised image*

*Fig 5.4 A noisy and denoised image of B2 bomber*

### 5.2. Filter performance quantification

The *PSNR* is the peak signal-to-noise ratio, in decibels, between an image and its filtered counterpart. This ratio is used as a quality measurement between the original and a filtered image. The higher the PSNR, the better the quality of the filter, or reconstructed image. The *Mean Square Error (MSE)* and the *Peak Signal to Noise Ratio (PSNR)* are the two error metrics used to compare image filtration quality. The *MSE* represents the cumulative squared error between the filtered and the original image, whereas *PSNR* represents a measure of the peak error. The lower the value of *MSE*, the lower the error. To compute the *PSNR*, the mean-squared error is calculated using the following equation:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[g(i,j) - f(i,j)]^2 \qquad 5.1$$

Where *M* X *N* is image size, g (i, j) is the Noisy image, f (I, j) is the filtered image. Then the PSNR is evaluated using equations 5.2. Figure 5.4 illustrates the noisy image and the result of filtration. Table 5.1 illustrates the results of filtering with different levels of noise.

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right) \qquad 5.2$$

Where 255 is the maximum intensity in an 8 bit image.

The computer simulation results show that the proposed algorithm is effective in extraction of speckle noise from images subjected to high noise levels. Although it does not give an account of the perceptible degradation to the cover-image, the PSNR is the most widely used metric in finding the quality of a filtered image.

### 5.3. Invariant Image moments.

Invariant moments are very sensitive to noise especially the moments of higher order. In the following procedure, invariant moments tests of a B2 bomber aircraft image with different noise levels was undertaken table 5.2. It is concluded that the noise levels affect the values of the invariant moments and therefore this may lead to erroneous results in the recognition process. Thus, the importance of preprocessing.

*Table 5.2 Invariant moments of an image with different levels of noise.*

| | $Ø_1$ | $Ø_2$ | $Ø_3$ | $Ø_4$ | $Ø_5$ | $Ø_6$ | $Ø_7$ |
|---|---|---|---|---|---|---|---|
| **B2 ORIGINAL SATELLITE IMAGE** | **6.61** | **14.05** | **15.24** | **17.45** | **33.94** | **24.67** | **39.26** |
| **B2 ( 10%NOISE)** | 6.55 | 13.91 | 15.03 | 17.24 | 33.53 | 24.40 | 39.24 |
| **B2 ( 20% NOISE)** | 6.54 | 13.88 | 14.96 | 17.17 | 33.39 | 24.32 | 38.61 |
| **B2** | 6.47 | 13.71 | 14.63 | 16.84 | 32.72 | 23.90 | 38.46 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **( 40% NOISE)** | | | | | | | |
| **B2** | 6.41 | 13.57 | 14.25 | 16.46 | 31.97 | 23.46 | 36.72 |
| **( 80% NOISE)** | | | | | | | |
| **B2** | Invariant moments cannot be evaluated | | | | | | |
| **( 90% NOISE)** | | | | | | | |
| **B2** | Invariant moments cannot be evaluated | | | | | | |
| **( 95% NOISE)** | | | | | | | |

Invariant moments for images above 80% noise level cannot be evaluated. Invariant moments exhibit translation, rotation and scaling invariance. Tests to show these properties were applied to a noisy image, and the results are shown in figure 5.5. Hu's moments which are calculated from the first 7 geometrical moments or up to the third order of moments were calculated. From the results illustrated in table 5.3, invariance of the moments can be readily deduced.

### 5.1.2 Invariance Tests.

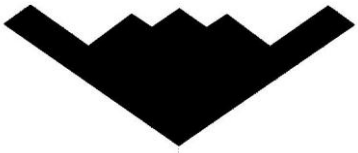


*Fig 5.5 a. Original image*

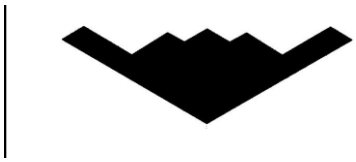


*Fig 5.5 b Translated image*

*Fig 5.5 c Scaled image*

*Fig 5.5 d  Rotated image (30$^o$ anticlockwise)*

*Fig 5.5 Invariance tests*

*Table 5.3 Invariance Tests*

| $\Phi_i$ | Original Image | Scaled Image | Rotated Image | Translated Image |
|---|---|---|---|---|
| $\Phi_1$ | 6.50 | 6.50 | 6.50 | 6.50 |
| $\Phi_2$ | 16.32 | 16.32 | 16.32 | 16.32 |
| $\Phi_3$ | 25.56 | 25.56 | 25.56 | 25.56 |
| $\Phi_4$ | 25.88 | 25.88 | 25.89 | 25.88 |
| $\Phi_5$ | 43.30 | 43.30 | 43.30 | 43.30 |
| $\Phi_6$ | 34.09 | 34.10 | 34.09 | 34.09 |
| $\Phi_7$ | 47.39 | 47.38 | 47.39 | 47.38 |

The conclusion deduced from table 5.3 is that there is moment invariance after geometrical transformations of scaling, rotation and translation. Invariant moments are computationally simple. However, they have several drawbacks: These are,

(iv) Information redundancy: since the basis is not orthogonal, these moments suffer from a high degree of information redundancy.

(v) Noise sensitivity: higher-order moments are very sensitive to noise.

(vi) Large variation in the dynamic range of values: since the basis involves powers of *p* and *q*, the moments computed may have large variation in the dynamic range of values for different orders. This may cause numerical instability when the image size is large.

### 5.3.1 Data Set

The following list is made up of six different classes of military aircrafts.

These are,

1. Fighter aircraft
2. Cargo aircraft
3. Bomber aircraft
4. Rotary wing aircraft
5. Reconnaissance aircraft
6. Training aircraft

There are hybrid aircrafts like the fighter bomber aircrafts. These are classified to the nearest class of the hybrid. This is a representative sampling of world military aircraft. The database is made up of American military aircrafts that is bombers, fighter aircrafts, cargo aircrafts, rotary aircrafts, trainer aircrafts, and reconnaissance planes and the rest of the world military aircrafts. Tables 5.4 to 5.9 exhibit these data.

### 5.3.2 The Training Set

*Table 5.4 Features extracted from fighter aircrafts*

| | Invariant Moments | | | | | | |
|---|---|---|---|---|---|---|---|
| **Aircraft Name** | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| A-6 intruder | 5.03 | 16.96 | 15.33 | 17.20 | 36.63 | 26.76 | 33.85 |
| Horten IX | 6.25 | 13.11 | 12.73 | 14.92 | 28.92 | 21.71 | 33.07 |
| Alpha jet | 4.97 | 15.32 | 12.47 | 14.67 | 28.43 | 22.56 | 29.99 |
| F14 tomcat | 5.74 | 13.96 | 12.96 | 15.17 | 29.47 | 22.40 | 30.36 |
| A-7 corsair II | 5.74 | 15.39 | 17.00 | 19.11 | 37.62 | 27.15 | 38.31 |
| F15 | 7.15 | 16.44 | 19.71 | 22.03 | 43.01 | 30.24 | 39.53 |
| F18 | 7.12 | 16.79 | 20.02 | 22.41 | 43.73 | 30.80 | 48.51 |
| F20 | 7.20 | 16.22 | 17.96 | 20.16 | 39.34 | 28.44 | 43.87 |
| F22 | 6.59 | 15.61 | 16.40 | 18.61 | 36.25 | 26.60 | 38.40 |
| Mig 21 | 7.03 | 15.20 | 19.12 | 21.39 | 41.77 | 28.99 | 47.00 |
| F35A | 7.18 | 17.54 | 19.70 | 21.89 | 42.82 | 30.83 | 45.04 |
| Mirage 2000 | 7.11 | 15.98 | 17.34 | 19.54 | 38.11 | 27.71 | 40.98 |
| F5 Freedom Fighter | 7.04 | 15.47 | 17.05 | 19.25 | 37.53 | 27.16 | 40.83 |
| A-10 Thunderbolt II | 5.66 | 17.92 | 19.37 | 21.06 | 41.53 | 30.33 | 42.21 |
| Mig23 | 7.19 | 17.79 | 19.52 | 21.71 | 42.44 | 30.76 | 44.87 |
| A-4 skyhawk | 5.12 | 15.99 | 11.96 | 14.12 | 27.40 | 22.35 | 28.38 |

*Table 5.5 Features extracted from cargo aircrafts*

| | Invariant Moments | | | | | | |
|---|---|---|---|---|---|---|---|
| **Aircraft Name** | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| C5 | 7.03 | 16.22 | 18.13 | 20.34 | 39.70 | 28.63 | 45.69 |
| Antonov | 7.26 | 20.06 | 22.45 | 24.65 | 48.31 | 34.83 | 52.89 |
| Rama | 7.28 | 17.63 | 19.74 | 21.93 | 42.89 | 30.91 | 48.26 |
| Aerostar | 7.27 | 19.38 | 21.74 | 23.95 | 46.90 | 33.79 | 51.52 |
| 123 RF | 7.18 | 16.86 | 18.74 | 20.92 | 40.87 | 29.52 | 44.34 |
| A40 | 7.27 | 19.79 | 22.49 | 24.73 | 48.44 | 34.77 | 51.68 |
| An condr | 7.23 | 19.81 | 22.09 | 24.28 | 47.56 | 34.32 | 51.51 |
| Il 2 | 7.27 | 28.64 | 27.52 | 29.16 | 57.59 | 43.48 | 59.82 |
| Il 14 | 7.18 | 16.59 | 18.47 | 20.67 | 40.37 | 29.14 | 44.64 |
| KC10 | 7.07 | 24.87 | 27.09 | 29.78 | 58.23 | 42.23 | 60.28 |
| C160 | 7.22 | 17.44 | 19.6 | 21.81 | 42.63 | 30.69 | 47.13 |
| C212 | 7.15 | 17.10 | 19.25 | 21.46 | 41.94 | 30.18 | 46.14 |
| Cn 235 | 7.27 | 19.63 | 22.51 | 24.71 | 48.42 | 34.53 | 52.32 |

*Table 5.6 Features extracted from bomber aircrafts*

| | **Invariant Moments** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Aircraft Name** | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| Sukhoi 25 | 7.25 | 18.59 | 21.5 | 23.74 | 46.47 | 33.04 | 52.01 |
| AMX | 7.22 | 16.55 | 18.40 | 20.59 | 40.21 | 29.04 | 45.30 |
| F16 | 6.58 | 15.49 | 17.00 | 19.20 | 37.44 | 27.13 | 39.56 |
| F5 tiger | 7.04 | 15.47 | 17.05 | 19.25 | 37.53 | 27.16 | 40.83 |
| MIG 31 | 7.08 | 16.43 | 17.74 | 19.94 | 38.91 | 28.33 | 41.19 |
| Rafale | 7.14 | 16.60 | 18.70 | 20.90 | 40.82 | 29.37 | 43.25 |
| Saab viggen | 7.05 | 16.01 | 17.19 | 19.38 | 37.80 | 27.57 | 40.12 |
| Sukhoi27 | 7.09 | 15.99 | 17.34 | 19.54 | 38.11 | 27.72 | 40.59 |
| YF117 | 7.00 | 16.06 | 17.98 | 20.22 | 39.45 | 28.43 | 41.73 |
| B2 | 6.61 | 14.05 | 15.25 | 17.45 | 33.95 | 24.68 | 39.26 |
| Delta wing | 7.04 | 16.56 | 18.53 | 20.73 | 40.48 | 29.18 | 45.06 |
| Q5 | 7.14 | 15.89 | 19.46 | 21.72 | 42.42 | 29.66 | 47.76 |
| J8 ii Shinyang | 7.0 | 14.91 | 15.94 | 18.13 | 35.31 | 25.78 | 39.65 |
| J9 | 7.11 | 15.80 | 17.35 | 19.55 | 38.14 | 27.63 | 43.45 |
| Sukhoi 30 | 7.22 | 17.41 | 20.42 | 22.70 | 44.37 | 31.40 | 48.65 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Super Galeb | 7.23 | 20.79 | 22.99 | 25.14 | 49.32 | 35.68 | 51.49 |
| Tornado | 7.21 | 17.51 | 20.56 | 22.84 | 44.65 | 31.59 | 51.64 |
| B52 | 7.17 | 18.49 | 21.23 | 23.41 | 45.84 | 32.65 | 49.07 |
| Yak 38 | 7.09 | 15.57 | 19.28 | 21.53 | 42.05 | 29.31 | 48.11 |
| J11 | 7.18 | 16.58 | 19.88 | 22.18 | 43.32 | 30.47 | 48.66 |
| Jh7 | 7.23 | 16.90 | 18.87 | 21.06 | 41.15 | 29.68 | 45.02 |
| Orao | 7.17 | 16.05 | 17.68 | 19.88 | 38.79 | 28.08 | 44.52 |
| IDF Taiwan | 7.16 | 16.25 | 18.09 | 20.28 | 39.60 | 28.58 | 42.80 |
| J10 | 7.04 | 15.89 | 17.67 | 19.87 | 38.77 | 27.99 | 41.79 |
| Jaguar | 7.18 | 16.54 | 19.85 | 22.15 | 43.27 | 30.42 | 50.03 |
| MCA | 7.20 | 16.14 | 17.73 | 19.93 | 38.88 | 28.17 | 42.77 |
| Mirage F1 | 7.17 | 16.29 | 19.68 | 21.95 | 42.88 | 30.09 | 49.93 |
| Sukhoi 7 | 7.03 | 15.51 | 19.20 | 21.53 | 42.02 | 29.29 | 46.98 |
| YAK 130 | 6.59 | 15.96 | 17.87 | 20.03 | 39.17 | 28.21 | 40.34 |
| X-45C | 7.00 | 17.30 | 19.74 | 21.75 | 42.62 | 30.42 | 44.29 |
| CAXF7IYK | 7.00 | 19.91 | 21.43 | 23.22 | 45.66 | 33.32 | 48.57 |

Table 5.7 F*eatures extracted from rotary aircrafts*

| | **Invariant Moments** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Aircraft Name** | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| Apache | 7.17 | 16.67 | 19.74 | 21.88 | 42.80 | 30.21 | 47.13 |
| Huey cobra | 7.05 | 15.20 | 19.11 | 21.28 | 41.59 | 28.88 | 47.62 |
| Eurocopter | 7.15 | 15.70 | 19.34 | 21.50 | 42.04 | 29.35 | 47.37 |
| Dauphin | 7.25 | 17.78 | 20.55 | 22.73 | 44.48 | 31.62 | 49.00 |
| Defender | 7.2680 | 18.73 | 21.35 | 23.50 | 46.04 | 32.87 | 50.72 |
| 532 Cougar | 7.22 | 16.67 | 19.81 | 21.97 | 42.99 | 30.31 | 49.15 |
| A129 Mangusta | 7.20 | 16.36 | 19.65 | 21.80 | 42.65 | 29.99 | 48.69 |
| Lynx | 7.23 | 16.72 | 19.88 | 22.05 | 43.13 | 30.41 | 48.44 |
| Mi4 Hound | 7.23 | 16.67 | 19.89 | 22.07 | 43.16 | 30.40 | 48.33 |
| Mi6 Hook | 7.20 | 16.51 | 19.74 | 21.91 | 42.84 | 30.16 | 47.93 |
| Mi8 Hip | 7.26 | 17.58 | 20.52 | 22.69 | 44.41 | 31.48 | 48.61 |
| Puma | 7.21 | 16.68 | 19.82 | 21.97 | 42.98 | 30.31 | 48.54 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SA 341 Gazelle | 7.26 | 18.36 | 20.96 | 23.10 | 45.24 | 32.28 | 48.03 |
| Yak Firebar | 7.19 | 16.56 | 19.84 | 22.07 | 43.14 | 30.35 | 47.56 |
| Ah 1 | 6.98 | 14.85 | 19.03 | 21.20 | 41.43 | 28.62 | 45.54 |
| AS 565 Pather | 4.78 | 11.16 | 12.53 | 14.10 | 29.02 | 20.63 | 27.61 |
| Ka 25 | 7.14 | 16.25 | 18.03 | 20.22 | 39.47 | 28.52 | 43.80 |
| Ka 52 Hokum | 7.21 | 17.29 | 20.30 | 22.48 | 43.99 | 31.13 | 47.88 |
| Mi 4 | 7.23 | 16.61 | 19.85 | 22.03 | 43.08 | 30.33 | 48.75 |
| Mi 6 | 7.24 | 17.43 | 20.38 | 22.54 | 44.12 | 31.25 | 50.24 |
| Mi 8 | 7.28 | 19.34 | 21.89 | 24.06 | 47.14 | 33.73 | 50.44 |
| Mi 24 | 7.23 | 17.37 | 20.28 | 22.44 | 43.92 | 31.13 | 47.61 |
| Mi 25 | 7.23 | 17.08 | 20.09 | 22.24 | 43.52 | 30.78 | 47.33 |
| Mi 26 | 7.24 | 17.43 | 0.38 | 2.54 | 4.11 | 1.25 | 50.24 |
| Mi 35 Havoc | 7.18 | 16.37 | 19.66 | 21.83 | 42.68 | 30.01 | 46.99 |
| Mi 35 Hind | 7.24 | 17.50 | 0.38 | 2.54 | 4.12 | 1.29 | 47.73 |
| SA 330 | 7.21 | 16.63 | 9.80 | 1.96 | 2.96 | 0.28 | 49.61 |
| SA 342 | 7.28 | 21.30 | 2.90 | 4.98 | 9.02 | 5.65 | 51.45 |
| Sa 360 | 4.81 | 11.26 | 3.10 | 4.50 | 8.57 | 1.00 | 29.40 |
| Ah 1 | 6.98 | 14.85 | 19.03 | 21.20 | 41.43 | 28.62 | 45.54 |

Table 5.8 F*eatures extracted from reconnaissance aircrafts*

| | Invariant Moments | | | | | | |
|---|---|---|---|---|---|---|---|
| **Aircraft Name** | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| Mitsubishi A5M | 7.05 | 16.15 | 18.22 | 20.48 | 39.95 | 28.73 | 44.10 |
| Northrop X-47B | 6.58 | 19.48 | 21.01 | 22.86 | 44.96 | 32.61 | 46.08 |
| Wirelizard | 7.21 | 17.09 | 19.07 | 21.28 | 41.57 | 29.99 | 47.94 |
| XL-RG | 7.27 | 17.61 | 19.73 | 21.92 | 42.86 | 30.89 | 47.18 |
| CAW4F62P | 7.19 | 16.27 | 19.63 | 21.86 | 42.71 | 29.99 | 51.30 |
| CA8Q39AI | 7.27 | 16.94 | 18.85 | 21.05 | 41.12 | 29.68 | 45.76 |
| CAORC6ZQ | 7.07 | 19.61 | 21.29 | 23.04 | 45.32 | 33.00 | 48.84 |
| A310-3 | 7.11 | 19.43 | 21.93 | 24.11 | 47.23 | 33.82 | 50.70 |
| An 71 | 7.21 | 16.88 | 18.85 | 21.05 | 41.12 | 29.66 | 46.26 |
| An 72 | 7.21 | 16.88 | 18.85 | 21.05 | 41.12 | 29.66 | 46.26 |
| EMB 145 | 6.67 | 15.58 | 16.56 | 18.75 | 36.55 | 26.73 | 39.48 |

Table 5.9 F*eatures extracted from training aircrafts*

| | Invariant Moments | | | | | | |
|---|---|---|---|---|---|---|---|
| **Aircraft Name** | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| A5 | 7.15 | 20.28 | 22.41 | 24.50 | 48.06 | 34.64 | 50.20 |
| RV-7A | 7.28 | 18.06 | 20.91 | 23.11 | 45.23 | 32.14 | 50.76 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| YAK 7 | 7.10 | 18.21 | 20.61 | 22.93 | 44.80 | 32.18 | 49.04 |
| VQ warbird | 7.18 | 17.97 | 20.43 | 22.72 | 44.40 | 31.86 | 48.87 |
| Buhligen | 7.11 | 17.64 | 20.10 | 22.45 | 43.83 | 31.42 | 49.28 |
| T 6 | 7.24 | 20.62 | 23.72 | 25.85 | 50.75 | 36.17 | 52.78 |
| Bellanca citabria | 7.25 | 17.06 | 19.02 | 21.22 | 41.47 | 29.92 | 49.00 |
| DRF | 7.29 | 18.90 | 21.67 | 23.87 | 46.75 | 33.32 | 50.91 |
| Casa_3 | 6.33 | 14.71 | 14.84 | 17.02 | 33.10 | 24.58 | 36.19 |

The training set is made up of samples from the data set for which the correct class label is known. Training is the process of estimating the classifier's parameters such as the means and variances of $P(x/C_i)$. Once the classifier has been trained then it can be used to label unknown data. These are indicated as unknown aircrafts on Table 5.10.

### 5.3.3 Test Set

The test set is used for the classifier evaluation. It contains data for which the class label is also known. A sample is presented to the classifier and compared to the classifier's output to the known correct label. Then the percentage of data that has been correctly labeled is determined. It is a statistical necessity that the training data and the test data are different. In this process the leave one out technique will be applied. One data vector will be left out and used as test sample while all the rest are used as training set. The process is repeated for as many tests as the data number. The classifier is thus evaluated.

## 5.4. BAYESIAN CLASSIFICATION

As presented in chapter 4, given an unknown pattern vector, the process is to compute a total of *W* decision functions and then assign the pattern to the class whose decision function yields the highest numerical value.

The Bayes classifier is designed for use when features are independent of one another within each class. It classifies data in three steps:

1. Training step: Using the training samples, the method estimates the parameters of a probability distribution, assuming features are conditionally independent in a given class.
2. Prediction step: For any unseen test sample, the method computes the posterior probability of that sample belonging to each class. The method then classifies the test sample according the largest posterior probability.
3. Identification step: Within a class the shortest distance from the data set is found and thus the sample identified.

The class event conditional probability independence assumption greatly simplifies the training step since one can estimate the one-dimensional class event conditional probability density for each feature individually. While the class-conditional independence between features is not true in general, research shows that this optimistic assumption works well in practice [46]. This assumption of class independence allows the Bayes classifier to better estimate the parameters required for accurate classification while using less training data than many other classifiers. This makes it particularly effective for datasets containing many predictors or features. The following table 5.10 shows two extracted features sets of unknown aircrafts that are to be classified.

*Table 5.10 Unknown Images Features*

| Aircraft Image | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Unknown Aircraft I | 6.19 | 16.11 | 21.76 | 23.28 | 45.85 | 33.31 | 46.41 |
| Unknown Aircraft II | 7.21 | 18.75 | 18.75 | 20.95 | 40.9 | 29.54 | 44.74 |

For a finite set of classes and an unknown feature vector *x* where *x* is a *d*-dimensional vector the conditional posterior probabilities for every class is calculated. This probabilities depend on the estimated values of the likelihood. The class for which the posteriori is maximum is chosen and this is the class to which the image whose feature vector produces this maximum belongs.

### 5.4.1 Multivariate Gaussian Classifier

Consider the 7-dimensional data *x* from the respective classes $C_i$ modelled using a multivariate Gaussian.

$$P(x/C) = P(x/\mu, \Sigma) = N(x; \mu, \Sigma)$$

$$= \frac{1}{(2\pi)^{\frac{d}{2}}\,\Sigma^{0.5}} exp\left(-\frac{1}{2}\,(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

5.3

The log Likelihood is given by.

$$LL(x/\mu, \Sigma) = lnP(x/\mu, \Sigma) = -\frac{d}{2}ln(2\pi) - \frac{1}{2}ln|\Sigma| - \frac{1}{2}\,(x-\mu)^T\Sigma^{-1}(x-\mu)$$

5.4

And the log Posterior Probability:

$$lnP(C/x) = -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) - \frac{1}{2}ln|\Sigma| + lnP(C) + const$$

5.5

Where μ is the mean of individual classes, ∑ is the class covariance and P(C) is the class Prior probability. It is also noted that $(x-\mu)^T\Sigma^{-1}(x-\mu)$ is the mahalanobis distance and d is the feature vector dimension.

The following table illustrates the world air forces fleet in year 2013 [57].

*Table 5.11a Aircraft type numbers*

| **Aircraft Type** | **Number** |
|---|---|
| Fighter aircraft fleet | 12568 |
| Bomber aircraft fleet | 3002 |
| Reconnaissance aircraft fleet | 1856 |
| Cargo aircraft fleet | 5145 |
| Helicopter aircraft fleet | 18686 |
| Training aircraft fleet | 10525 |
| **TOTAL** | **51782** |

*Table 5.11b Class prior probabilities.*

| Class Label | Class Prior Probabilities |
|---|---|
| Fighter | 0.2427 |
| Cargo | 0.0994 |
| Bomber | 0.0580 |
| Rotary | 0.3609 |

| | |
|---|---|
| Reconnaissance | 0.0358 |
| Training | 0.2032 |
| TOTAL | 1 |

*Table 5.11 Class prior probabilities*

The Bayes Gauss algorithm as illustrated in Appendix 10.2 shows that max-posterior for unknown aircraft I is a Fighter class aircraft and unknown aircraft II is a Rotary wing class aircraft.

### 5.4.2 Identification

The method used is intra-class distance evaluation. Having known that the aircraft is in fighter class the next step is to get the distance of all the type vectors in that class from the unknown plane. The one with the shortest distance will be the aircraft type. Table 5.12 illustrates the distances of the fighter class aircraft from the unknown aircraft.

*Table 5.12 Fighter class/unknown aircraft I distances.*

| Aircraft Type | Distance from unknown aircraft |
|---|---|
| Mig23 | 5.64 |
| A-4 skyhawk | 31.09 |
| A-10 A ThunderboltII | 7.71 |
| Alpha jet | 29.18 |
| F-5 freedom fighter | 34.77 |

| | |
|---|---|
| | 28.16 |
| F14 tomcat | |
| | 12.46 |
| Mirage 2000 | |
| | 5.83 |
| F15 | |
| F35A | 5.14 |

*Table 5.13* Rotary wing/unknown aircraft II distances.

| Aircraft Type | Distance from unknown aircraft |
|---|---|
| | 3.99 |
| Apache | |
| Huey cobra | 4.70 |
| Eurocopter | 4.27 |
| Dauphin | 6.53 |
| Defender | 9.30 |

Therefore from the evaluated distances the unknown aircraft I type corresponds to type F35A while unknown aircraft II is like type Apache helicopter.

Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from dimensionality, such as the need for data sets that

scale exponentially with the number of features. Like all probabilistic classifiers under the MAP decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; hence class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying probability model [2].

### 5.5. Vector Quantization

#### 5.5.1. Clustering

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to *k*, *k*-means clustering gives a formal definition as an optimization problem. That is to find the *k* cluster centers and to assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

Table 5.14 illustrates vector features of 50 aircrafts that are used as clustering sample. They represent a sample of all aircraft classes.

*Table 5.14 Invariant moments for training patterns.*

| | | Moments ($\Phi_{i)}$ | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Aircraft Image | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ |
| 1 | Apache | 7.17 | 16.67 | 19.74 | 21.88 | 42.80 | 30.21 | 47.13 |
| 2 | A5 | 7.15 | 20.28 | 22.41 | 24.50 | 48.06 | 34.64 | 50.20 |
| 3 | C5 | 7.03 | 16.22 | 18.13 | 20.34 | 39.70 | 28.63 | 45.69 |
| 4 | Mig23 | 7.19 | 17.79 | 19.52 | 21.71 | 42.44 | 30.76 | 44.87 |
| 5 | A-4 skyhawk | 5.12 | 15.99 | 11.96 | 14.12 | 27.40 | 22.35 | 28.38 |
| 6 | A-10 A ThunderboltII | 5.66 | 17.92 | 19.37 | 21.06 | 41.53 | 30.33 | 42.21 |
| 7 | A-6 intruder | 5.03 | 16.96 | 15.33 | 17.20 | 36.63 | 26.76 | 33.85 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | A-7 corsairII | 5.74 | 15.39 | 17.00 | 19.11 | 37.62 | 27.15 | 38.31 |
| 9 | Alpha jet | 4.97 | 15.32 | 12.47 | 14.67 | 28.43 | 22.56 | 29.99 |
| 10 | AMX | 7.22 | 16.55 | 18.40 | 20.59 | 40.21 | 29.04 | 45.30 |
| 11 | Image 12 | 7.21 | 16.33 | 18.04 | 20.23 | 39.49 | 28.57 | 45.07 |
| 12 | F-5 freedom fighter | 5.15 | 12.27 | 10.92 | 13.19 | 25.46 | 20.04 | 26.86 |
| 13 | F14 tomcat | 5.74 | 13.96 | 12.96 | 15.17 | 29.47 | 22.40 | 30.36 |
| 14 | F16 | 6.58 | 15.49 | 17.00 | 19.20 | 37.44 | 27.13 | 39.56 |
| 15 | F5 tiger | 7.04 | 15.47 | 17.05 | 19.25 | 37.53 | 27.16 | 40.83 |
| 16 | Horten IX | 6.25 | 13.11 | 12.73 | 14.92 | 28.92 | 21.71 | 33.07 |
| 17 | MIG 21 | 7.01 | 15.10 | 16.32 | 18.51 | 36.06 | 26.25 | 40.68 |
| 18 | MIG 31 | 7.08 | 16.43 | 17.74 | 19.94 | 38.91 | 28.33 | 41.19 |
| 19 | Mitsubishi A5M | 7.05 | 16.15 | 18.22 | 20.48 | 39.95 | 28.73 | 44.10 |
| 20 | Northrop X-47B | 6.58 | 19.48 | 21.01 | 22.86 | 44.96 | 32.61 | 46.08 |
| 21 | Rafale | 7.14 | 16.60 | 18.70 | 20.90 | 40.82 | 29.37 | 43.25 |
| 22 | Saab viggen | 7.05 | 16.01 | 17.19 | 19.38 | 37.80 | 27.57 | 40.12 |
| 23 | Sukhoi27 | 7.09 | 15.99 | 17.34 | 19.54 | 38.11 | 27.72 | 40.59 |
| 24 | X-45C | 7.00 | 17.30 | 19.74 | 21.75 | 42.62 | 30.42 | 44.29 |
| 25 | YAK 130 | 6.59 | 15.96 | 17.87 | 20.03 | 39.17 | 28.21 | 40.34 |
| 26 | YF117 | 7.00 | 16.06 | 17.98 | 20.22 | 39.45 | 28.43 | 41.73 |
| 27 | RV-7A | 7.28 | 18.06 | 20.91 | 23.11 | 45.23 | 32.14 | 50.76 |
| 28 | YAK 7 | 7.10 | 18.21 | 20.61 | 22.93 | 44.80 | 32.18 | 49.04 |
| 29 | VQ warbird | 7.18 | 17.97 | 20.43 | 22.72 | 44.40 | 31.86 | 48.87 |
| 30 | M5 | 7.11 | 15.81 | 17.38 | 19.59 | 38.20 | 27.67 | 47.36 |
| 31 | Antonov | 7.26 | 20.06 | 22.45 | 24.65 | 48.31 | 34.83 | 52.89 |
| 32 | Rama | 7.28 | 17.63 | 19.74 | 21.93 | 42.89 | 30.91 | 48.26 |
| 33 | Mirage 2000 | 7.11 | 15.98 | 17.34 | 19.54 | 38.11 | 27.71 | 40.98 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 34 | Buhligen | 7.11 | 17.64 | 20.10 | 22.45 | 43.83 | 31.42 | 49.28 |
| 35 | Bellanca citabria | 7.25 | 17.06 | 19.02 | 21.22 | 41.47 | 29.92 | 49.00 |
| 36 | Drf lufftrettung | 7.21 | 16.33 | 18.04 | 20.23 | 39.49 | 28.57 | 45.07 |
| 37 | DRF | 7.29 | 18.90 | 21.67 | 23.87 | 46.75 | 33.32 | 50.91 |
| 38 | A5 | 7.03 | 16.22 | 18.13 | 20.34 | 39.70 | 28.63 | 45.69 |
| 39 | Aerostar | 7.27 | 19.38 | 21.74 | 23.95 | 46.90 | 33.79 | 51.52 |
| 40 | Wirelizard | 7.21 | 17.09 | 19.07 | 21.28 | 41.57 | 29.99 | 47.94 |
| 41 | XL-RG | 7.27 | 17.61 | 19.73 | 21.92 | 42.86 | 30.89 | 47.18 |
| 42 | 123 RF | 7.18 | 16.86 | 18.74 | 20.92 | 40.87 | 29.52 | 44.34 |
| 43 | CAW4F62P | 7.19 | 16.27 | 19.63 | 21.86 | 42.71 | 29.99 | 51.30 |
| 44 | CA8Q39AI | 7.27 | 16.94 | 18.85 | 21.05 | 41.12 | 29.68 | 45.76 |
| 45 | CAXF7IYK | 7.00 | 19.91 | 21.43 | 23.22 | 45.66 | 33.32 | 48.57 |
| 46 | CAORC6ZQ | 7.07 | 19.61 | 21.29 | 23.04 | 45.32 | 33.00 | 48.84 |
| 47 | B2 | 6.61 | 14.05 | 15.25 | 17.45 | 33.95 | 24.68 | 39.26 |
| 48 | Delta wing | 7.04 | 16.56 | 18.53 | 20.73 | 40.48 | 29.18 | 45.06 |
| 49 | F15 | 7.15 | 16.44 | 19.71 | 22.03 | 43.01 | 30.24 | 49.53 |
| 50 | F35A | 7.18 | 17.54 | 19.70 | 21.89 | 42.82 | 30.83 | 45.04 |
| | Unknown aircraft I | 6.19 | 16.11 | 21.76 | 23.28 | 45.85 | 33.31 | 46.41 |

In table 5.15, starting with 5 clusters, the number of clusters are gradually increased to see if k-means can find a better grouping of the data. The total sum of distances decreases at each iteration as k-means reassigns points between clusters and recomputes cluster centroids. In this case, the minimum was reached after five iterations. A silhouette plot for this solution indicates that these five clusters give a better separation than one, two or three. However there would not be any need of a higher number of clusters than five as indicated by the 10 clusters. This is shown on tables 5.16 and 5.18.

Table 5.15 a k-means clustering

| | Aircraft Image | class in 5 clusters | class in 10 clusters |
|---|---|---|---|
| 1 | Apache | 5 | 9 |
| 2 | A5 | 3 | 8 |
| 3 | C5 | 1 | 7 |
| 4 | Mig23 | 1 | 4 |
| 5 | A-4 skyhawk | 2 | 3 |
| 6 | A-10 A ThunderboltII | 1 | 4 |
| 7 | A-6 intruder | 4 | 5 |
| 8 | A-7 corsairII | 4 | 10 |
| 9 | Alpha jet | 2 | 3 |
| 10 | AMX | 1 | 7 |
| 11 | Image 12 | 1 | 7 |
| 12 | F-5 freedom fighter | 2 | 3 |
| 13 | F14 tomcat | 2 | 3 |
| 14 | F16 | 4 | 10 |
| 15 | F5 tiger | 4 | 10 |
| 16 | Horten IX | 2 | 3 |
| 17 | MIG 21 | 4 | 10 |
| 18 | MIG 31 | 4 | 10 |
| 19 | Mitsubishi A5M | 1 | 7 |
| 20 | Northrop X-47B | 3 | 6 |
| 21 | Rafale | 1 | 1 |
| 22 | Saab viggen | 4 | 10 |
| 23 | Sukhoi27 | 4 | 10 |
| 24 | X-45C | 1 | 4 |

| | | | |
|---|---|---|---|
| 25 | YAK 130 | 4 | 10 |
| 26 | YF117 | 4 | 10 |
| 27 | RV-7A | 3 | 6 |
| 28 | YAK 7 | 3 | 6 |
| 29 | VQ warbird | 5 | 6 |
| 30 | M5 | 1 | 2 |
| 31 | Antonov | 3 | 8 |
| 32 | Rama | 5 | 9 |
| 33 | Mirage 2000 | 4 | 10 |
| 34 | Buhligen | 5 | 9 |
| 35 | Bellanca citabria | 5 | 9 |
| 36 | Drf lufftrettung | 1 | 7 |
| 37 | DRF | 3 | 8 |
| 38 | A5 | 1 | 7 |
| 39 | Aerostar | 3 | 8 |
| 40 | Wirelizard | 5 | 9 |
| 41 | XL-RG | 5 | 9 |
| 42 | 123 RF | 1 | 1 |
| 43 | CAW4F62P | 5 | 9 |
| 44 | CA8Q39AI | 1 | 1 |
| 45 | CAXF7IYK | 3 | 6 |
| 46 | CAORC6ZQ | 3 | 6 |
| 47 | B2 | 4 | 5 |
| 48 | Delta wing | 1 | 1 |
| 49 | F15 | 5 | 9 |
| 50 | F35A | 1 | 4 |

Table 5.15 b k-means clustering for unknown aircrafts

| Aircraft Image | class in 5 clusters | class in 10 clusters |
|---|---|---|
| Unknown aircraft I | 1 | 4 |
| Unknown aircraft II | 5 | 9 |

*The k*-means clustering is a partitioning method, *k*-means treats each observation in the data as an object having a location in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible as shown in the data scatter figure 5.6. Each cluster in the partition is defined by its member objects and by its centroid, or center.

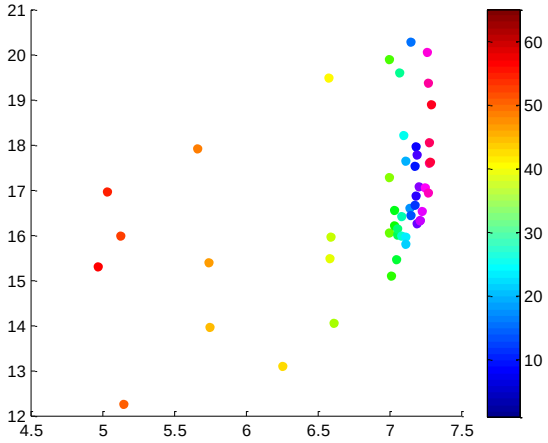



*Fig 5.6 Data scatter*

The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized, *k*-means uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well-separated as possible as illustrated in Table 5.16.

Table 5.16 Cluster analysis

| CLUSTE R NAME | Feature vectors of Centroids | Region al |
|---|---|---|

Data

| | $\Phi_1$ | $\Phi_2$ | $\Phi_3$ | $\Phi_4$ | $\Phi_5$ | $\Phi_6$ | $\Phi_7$ | |
|---|---|---|---|---|---|---|---|---|
| $C_1$ | .02 | | L.53 | 3.54 | 5.18 | 3.31 | ).52 | 16 |
| $C_2$ | .04 | | 3.63 | ).79 | ).63 | ).33 | 1.87 | 5 |
| $C_3$ | .20 | | ).69 | L.92 | 2.84 | ).60 | 3.72 | 9 |
| $C_4$ | .66 | | 5.95 | ).11 | 7.57 | 7.26 | ).79 | 12 |
| $C_5$ | .45 | | 2.21 | 1.42 | 7.94 | L.81 | ).73 | 10 |

From the above results it is deduced that while the more precise Bayesian classifier classifies the given samples as to F35A and Apache aircraft respectively, the Vector Quantization clusters the unknown aircrafts in the same clusters with F35A and Apache aircraft respectively figure 5.7 illustrates the F35A aircraft.

*Fig 5.7 The F35A aircraft*

## 5.6. Optimization

### 5.6.1 Minimizing the Classification Error Probability

The Bayesian classifier is optimal with respect to minimizing the classification error probability. Moving the threshold away from $x_0$, in Figure 4.1, always increases the corresponding area under the curves. Let $R_1$ be the region of the feature space in which we decide in favour of $\omega_1$ and $R_2$ be the corresponding region for $\omega_2$. Then an error is made if x ε $R_1$, although it belongs to $\omega_2$ or if x ε $R_2$, although it belongs to $\omega_1$. That is,

$$P_e = P(x \varepsilon R_2, \omega_1) + P(x \varepsilon R_1, \omega_2) \quad 5.6$$

Where $P_e$ is the joint probability of two events. This becomes,

$$P_e = P(x\varepsilon R_2/\omega_1)P(\omega_1) + P(x\varepsilon R_1/\omega_2)P(\omega_2) \quad 5.7$$

$$=P(\omega_1)\int_{R_2} p\,(x/\omega_1)\,dx + P(\omega_2)\int_{R_1} p\,(x/\omega_2)\,dx \quad 5.8$$

Or using the Bayes rule

$$P_e = \int_{R_2} P(\omega_1/x)p\,(x)\,dx \int_{R_1} P(\omega_2/x)p\,(x)\,dx \quad 5.9$$

It is now easy to see that the error is minimized if the partitioning regions $R_1$ and $R_2$ of the feature space are chosen so that

$$R_1: P(\omega_1/x) > P(\omega_2/x) \qquad R_2: P(\omega_2/x) > P(\omega_1/x) \quad 5.10$$

Indeed, since the union of the regions $R_1$ and $R_2$ covers all the space, from the definition of a probability density function we have:

$$\int_{R_1} P(\omega_1/x)p\,(x)\,dx + \int_{R_2} P(\omega_1/x)p\,(x)\,dx = P(\omega_1) \quad 5.11$$

Therefore

$$P_e = P(\omega_1) - \int_{R_1}(P(\omega_1/x) - P(\omega_2/x))p(x)dx \quad 5.12$$

This suggests that the probability of error is minimized if $R_1$ is the region of space in which $P(\omega_1/x) > P(\omega_2/x)$. Then, $R_2$ becomes the region where the reverse is true. Generalizations to the multiclass case are straightforward. In a classification task with M classes, $\omega_1, \omega_2, \omega_3, \omega_4 \ldots\ldots \omega_M$ an unknown pattern, represented by the feature vector x, is assigned to class $\omega_i$ if

$$P\,(\omega_i|x) > P\,(\omega_j|x) \quad 5.13$$

It turns out that such a choice also minimises the classification error probability.

**5.6.2. Creating Clusters and Determining Separation** To show how well-separated the resulting clusters are, a silhouette is plotted using the cluster indices output from *k*-means. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighbouring clusters. This measure ranges from +1, indicating points that are very distant from neighbouring clusters, through 0, indicating points that are not distinctly in one cluster or another, to -1, indicating points that are obviously assigned to the wrong cluster. This can be observed in figures 5.8 and 5.9.

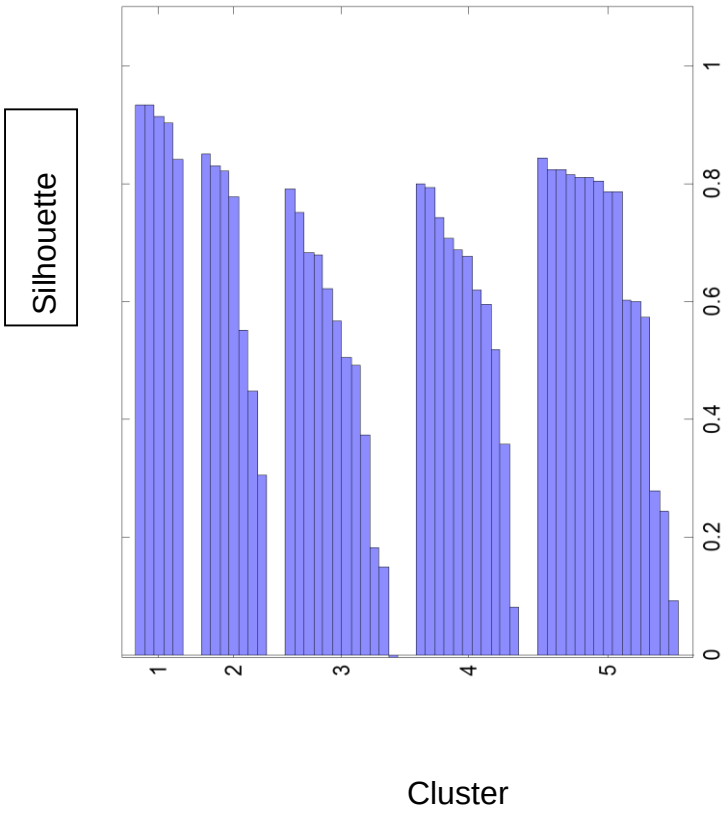



*Fig 5.8 Five Clusters silhouette*

From the silhouette plot, most points in the clusters have a large silhouette value, greater than 0.6, indicating that the cluster is somewhat separated from neighbouring clusters. However, the third and fifth clusters contain few points with low silhouette values. This means they are near the borders of other clusters.

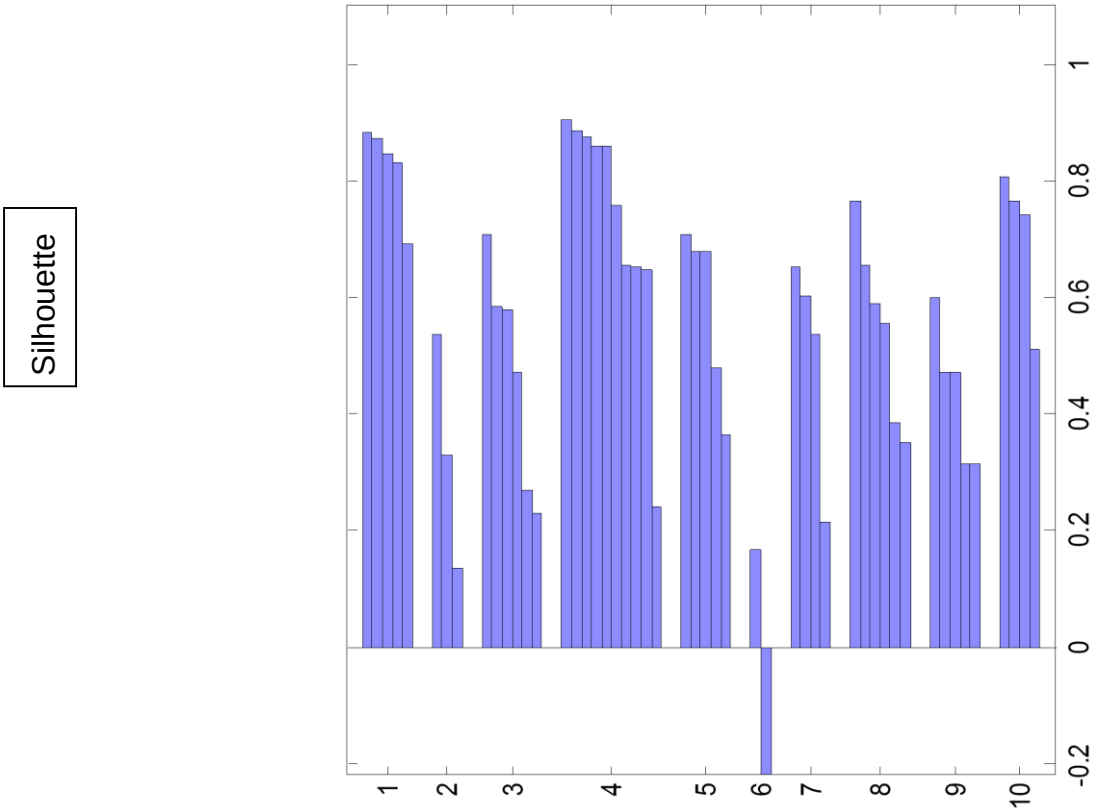



*Fig 5.9 Ten cluster silhouette*

The silhouette figure 5.9 indicates that the output in cluster 6 is assigned to very near the border on the other side of the cluster where it is situated.

**5.63. Determining the Correct Number of Clusters** The number of clusters is increased to see if *k*-means can find a better grouping of the data, by minimizing the sum of the distances between the clusters. Information about each iteration is given in table 5.17

*Table 5.17 Distance minimization of 10 clusters.*

| Iteration | Distance sum |
|---|---|
| 1 | 862.23 |

| | |
|---|---|
| 2 | 595.66 |
| 3 | 543.87 |
| 4 | 441.75 |
| 5 | 337.81 |
| 6 | 337.81 |

*Table 5.18 Distance minimization of 5 clusters.*

| Iteration | Distance sum |
|---|---|
| 1 | 223.49 |
| 2 | 185.66 |
| 3 | 164.9 |
| 4 | 158.13 |
| 5 | 153.76 |
| 6 | 153.76 |

The tables above show that after 6 iterations the total sum of distances from the centroids is 337.807 if ten cluster are used and the mean separation is 0.6192. At the same time if five clusters are used and after 5 iterations the total sum of distances from the centroids is 153.757 while the mean separation is 0.6419.

It can be seen that with 10 clusters minimization of the sums and thus of the centroids is reached faster after only 5 iterations.

The total sum of distances is equal to 153.757 and the mean separation is equal to 0.6192.In conclusion it is seen that 5 clusters give a better separation than 10 clusters. It would therefore be unnecessary to use more than 5 clusters for the classification.

### 5.6.4. Performance evaluation in respect to the probability of classification error.

Error Counting Approach:

Consider an *M* class classification task. The objective is to estimate the classification error probability by testing the "correct/false" response of an independently designed classifier using a finite set of *N* test feature vectors. In this case every class is a test feature vector. Let $N_i$ be the vectors in each class with

$$\sum_{i}^{M} N_i = N$$

5.14

And $P_i$ the corresponding error probability for class $\omega_i$. Assuming independence among the feature vectors, the probability that $k_i$ vectors from class $\omega_i$ being misclassified is given by the binomial distribution [5].

$$P(k_i misclassified) = \binom{N_i}{k_i} P_i^{k_i}(1 - P_i)^{N_i - k_i}$$

5.15

In our case the probabilities $P_i$ are not known. An estimate $\widehat{P_i}$ results if we maximize with respect to $P_i$. Differentiating and equating to zero results in the estimate,

$$\widehat{P_i} = \frac{k_i}{N_i}$$

5.16

Thus, the total error probability estimate is given by:

$$\widehat{P_i} = \sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{P}(\omega_\mathrm{i}) \frac{k_i}{N_i}$$

5.17

Where $P(\omega_i)$ is the occurrence probability of class $\omega_i$. And $\widehat{P_i}$ is an unbiased estimate of the true error probability. The minimum size of the test data set, *N*, is derived in terms of the true error probability *P* of the already designed classifier. Thus, if very small data sets are used for testing the performance of a classifier, the resulting estimate may not be reliable.

### 5.6.5. Data Set Finite Size Exploitation:

**Resubstitution Method:** The same data set is used, first for training and then for testing. One need not go into mathematical details in order to see that such a procedure is not very fair. Indeed, this is justified by the mathematical analysis. The performance of this method were analyzed using normal distributions. The analysis results show that this method provides an optimistic estimate of the true error probability. The amount of bias of the resubstitution estimate is a function of the ratio *N/l*, that is, the data set size and the dimension of the feature space. Furthermore, the variance of the estimate is inversely proportional to the data set size *N*. In words, in order to obtain a reasonably good estimate, *N* as well as the ratio $\frac{N}{l}$ must be large enough. The results from the analysis and the related simulations show that $\frac{N}{l}$ should be at least three and that an upper bound of the variance is $\frac{1}{8}N$. The larger the ratio $\frac{N}{l}$ the more comfortable the analysis [5].

In this thesis the data size N=108

Data dimension l=7

Therefore, *N/l*=15.7

Upper bound of the variance = 1/8*N*

= 13.5

### 5.6.6. Holdout Method:

The available data set is divided into two subsets, one for training and one for testing. The major drawback of this technique is that it reduces the size for both the training and the testing data. Another problem is to decide how many of the *N* available data will be allocated to the training set and how many to the test set. A finite data set introduces an excess mean error and a variance around it, as different data sets, of the same size, are used for the design. Both of these quantities depend on the size of the training set. The classification error probability of a classifier, designed using a finite training data set, *N*, is always higher than the corresponding asymptotic error probability ($N \rightarrow \infty$). This excess error decreases as *N* increases. On the other hand, the variance of the error counting depends on the size of the test set, and for small test data sets the estimates can be unreliable. Efforts made to optimize the respective sizes of the two sets have not yet led to practical results [5].

**5.6.7. Leave-One-Out Method:** This method alleviates the lack of independence between the training and test sets in the resubstitution method and at the same time frees itself from the dilemma associated with the holdout method. The training is performed using *N*-1 samples, and the test is carried out using the excluded sample. If this is misclassified, an error is counted. This is repeated *N* times, each time excluding a different sample. The total number of errors leads to the estimation of the classification error probability. Thus, training is achieved using, basically, all samples, and at the same time independence between training and test sets is maintained. The major disadvantage of the technique is its high computational complexity. For certain types of classifiers (i.e., linear or quadratic) it turns out that a simple relation exists between the leave-one-out and the resubstitution. Thus, in such cases the former estimate is obtained using the latter method with some computationally simple modifications. The holdout error estimate, for a Bayesian classifier, is an upper bound of the true Bayesian error. In contrast, the resubstitution error estimate is a lower bound of the Bayesian error.

i. $P_e^N$ Denotes the classification error probability for a classifier designed using a finite set of N training samples
ii. $P_e^{-N}$ Denotes the average E [$P_e^N$ ] over all possible training sets of size *N*.
iii. $P_e$ is the average asymptotic error as $N \rightarrow \infty$.

The holdout and leave-one-out methods (for statistically independent samples) provide an unbiased estimate of $P_e^{-N}$. In contrast, the resubstitution method provides a biased (underestimated) estimate of $P_e^{-N}$ as the data size *N* increases, both curves tend to approach the asymptotic $P_e$. As illustrated on figure 5.10

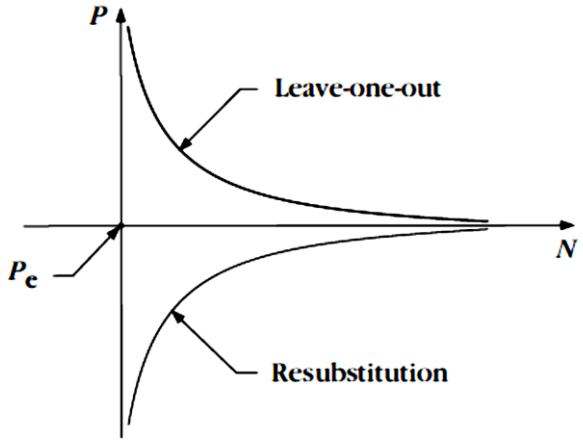



*Fig 5.10 Leave-One-Out method [25]*

### 5.6.8. Error Matrix, Recall and Precision

An error Matrix is a visual performance assessment of a classification algorithm in the form of a table layout or matrix. Each column of the matrix represents predicted classifications and each row represents actual defined classifications. This representation is a useful way to help evaluate a classifier model. A well behaved model should produce consistent percentage correctness numbers for accuracy, recall, precision and an F measure. If it does not, there is cause to further evaluate the data used to train the model and the data used to test the model. This is a helpful way to evaluate the classifier. The classification system has been trained to distinguish between aircrafts, an error table will summarize the results of testing the algorithm. Using the sample of 108 aircrafts the error table is as illustrated in table 5.19.

Table 5.19 Error table

| | Aircraft | Identified as |
|---|---|---|
| 1 | Apache | Apache |
| 2 | A5 | A5 |
| 3 | C5 | C5 |
| 4 | Mig23 | Mig23 |
| 5 | A-4 skyhawk | A-4 skyhawk |

| | | |
|---|---|---|
| 6 | A-10 A ThunderboltII | A-10 A ThunderboltII |
| 7 | A-6 intruder | A-6 intruder |
| 8 | A-7 corsairII | YAK 7 |
| 9 | Alpha jet | Alpha jet |
| 10 | AMX | AMX |
| 11 | Image 12 | Image 12 |
| 12 | F-5 freedom fighter | F-5 freedom fighter |
| 13 | F14 tomcat | F14 tomcat |
| 14 | F16 | F16 |
| 15 | F5 tiger | F5 tiger |
| 16 | Horten IX | Horten IX |
| 17 | MIG 21 | MIG 21 |
| 18 | MIG 31 | MIG 31 |
| 19 | Mitsubishi A5M | Mitsubishi A5M |
| 20 | Northrop X-47B | Northrop X-47B |
| 21 | Rafale | Rafale |
| 22 | Saab viggen | Saab viggen |
| 23 | Sukhoi27 | Sukhoi27 |
| 24 | X-45C | X-45C |
| 25 | YAK 130 | YAK 130 |
| 26 | YF117 | YF117 |
| 27 | RV-7A | RV-7A |
| 28 | YAK 7 | A-7 corsairII |
| 29 | VQ warbird | VQ warbird |
| 30 | M5 | M5 |
| 31 | Antonov | Antonov |
| 32 | Rama | Rama |
| 33 | Mirage 2000 | Mirage 2000 |

| | | |
|---|---|---|
| 34 | Buhligen | Buhligen |
| 35 | Bellanca citabria | Bellanca citabria |
| 36 | Drf lufftrettung | Drf lufftrettung |
| 37 | DRF | DRF |
| 38 | A5 | A5 |
| 39 | Aerostar | Aerostar |
| 40 | Wirelizard | Wirelizard |
| 41 | XL-RG | XL-RG |
| 42 | 123 RF | 123 RF |
| 43 | CAW4F62P | CAW4F62P |
| 44 | CA8Q39AI | CA8Q39AI |
| 45 | CAXF7IYK | CAXF7IYK |
| 46 | CAORC6ZQ | CAORC6ZQ |
| 47 | B2 | B2 |
| 48 | Delta wing | Delta wing |
| 49 | F15 | F15 |
| 50 | F35A | F35A |

Columns represent predictions made by the algorithm. In the first row, one aircraft was classified correctly as Apache. The 8$^{th}$ and 28$^{th}$ aircrafts were incorrectly classified as the 28$^{th}$ and 8$^{th}$ aircrafts respectively.The following two aircrafts figure 5.11(a) and 5.11(b).

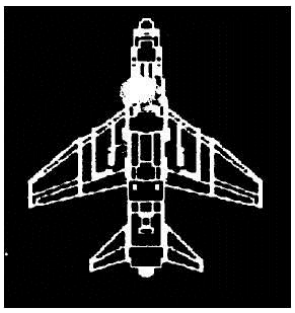


Fig 5.11 (a) Yak 7 aircraft (b) A7 Corsair II

Accuracy is defined as the correct classifications divided by all classifications. From the above table,

True Predictions = 48

False Predictions = 2

Accuracy = 0.96

Recall is defined as the number of correct classifications penalized by the number of missed items.

Recall = 48/50

=0.96

Precision is defined as the correct classifications penalized by the number of incorrect classifications.

Precision = 0.96

**F Measure (F):**

F measure is a derived effectiveness measurement. The resultant value is interpreted as a weighted average of the precision and recall. The best value is 1 and the worst is 0.

F = 2((precision*recall) / (precision+recall)).

F = 2((0.96*0.96)/ (0.96+0.96)).

F = 0.96

All the data is reasonably balanced across the table showing a majority of correct classifications.

# CHAPTER 6

# CONCLUSION AND RECOMMENDATIONS

**6.1. The Classifier** Statistical pattern recognition is a term used to cover all stages of an investigation from problem formulation and data collection through to discrimination and classification, assessment of results and interpretation. The topic of invariant moments feature extraction classification was discussed showing that the application based on moment's features extraction separate from Bayesian classifiers has been successful. As previously mentioned work carried out by A. Kadir et al [14] successfully identified plant leaves using leaf shape features. Likewise L. Keyes and A.C. Winstanley [15] in a paper titled "Topographic object recognition through shape" successfully presented feature coding and recognition of topographic data. Tables 5.4 illustrates the invariant moments of the fighter class aircrafts. Table 5.10 illustrates the invariant features of unknown aircrafts and it would have sufficed to compare the distance metrics between the unknown aircrafts and all the other classes of aircrafts for classification. However the two applications have not yet previously been combined to see how successful they could be. The main topic concerns classifier design: given a training set of patterns of known class, a classifier has been designed that is optimal for the expected operating conditions (the test conditions). The classifier is not too complex (there are too many free parameters) so as not to model noise in the design set i.e. over-fitting. The classifier is complex enough, to capture structure in the data. Several training factors could influence the accuracy of the Bayesian classifier and the training images could themselves be a source of uncertainty.

**6.2. Misrecognition** An error or misrecognition occurs when the classifier assigns a pattern to one class when it actually belongs to another. It is the uncertain classifications which mainly contribute to the error rate. Therefore, rejecting a pattern (withholding a decision) may lead to a reduction in the error rate. This rejected pattern may be discarded, or set aside until further information allows a decision to be made. Although the option to reject may alleviate or remove the problem of a high misrecognition rate, some otherwise correct classifications are also converted into rejects. Therefore trade-offs should be considered between error rate and reject rate. A clustering algorithm can be employed to reveal the groups in which feature vectors are clustered in the N-dimensional feature space. From the results it can be seen that the clustering vector quantization method can be used for a fast hand classification that is not very precise and the Bayesian classification can be used for a precise classification. However the precision of the VQ method can be augmented by increasing the number of codewords in the Voronoi diagram. These is clearly deduced from the following results.

### 6.3 Recommendations

Based on the results presented here, success can be claimed in demonstrating shape classification by the recognition and feature coding of satellite aircraft images using a Bayesian approach and invariant moments.

However, to further investigate and develop this project the following recommendations for the future extension of this research are made:

(i) Of most obvious need of improvement is the graphics resolution of the satellite images that are to be used.

(ii) More trials could be performed on more data this would generate fuller results for supervised learning using the statistical analysis.

(iii) Extend further the types of feature descriptors utilized.

(iv) Finally, the aircraft could be represented as a three dimensional model, and manipulated through homogeneous transformations. This would allow the aircraft to be scaled, translated, and rotated, and then displayed in any of an infinite number of aspects.

# REFERENCES


1 M.K. Hu. “Visual pattern recognition by moment invariants,” *IRE Trans. Information Theory*, vol. IT-8, pp. 179- 187, 1962.

2 R. O. Duda, P. E. Hart and D.G. Stork. *Pattern Classification,* US: John Wiley and Sons, 2007.

3 W.K. Pratt. *Digital Image Processing,* California, and U.S.A.: Wiley Interscience, 2007.

4 S. Y. Chen, J. Zhang, Q. Guan and S. Liu. “Detection and amendment of shape distortions based on moment invariants for active shape models”. *IET Image Processing*, vol. 5, no. 3, pp. 273–285, 2011.

5 S.Theodoridis, K. Koutroumbas. *Pattern recognition*. US: Elsevier, 2009.

6 W. Cao, S. Meng. “Imaging systems and Recognition Techniques”. *IEEE International Workshop*, IST.2009.5071625, pp 164-167, Shenzhen, 2009.

7 D. Maltoni, D. Maio, A. K. Jain and S. Prabhakar. *Handbook of Fingerprint recognition*. US: Springer, 2009.

8 A. K. Jain, *Fundamentals of Digital Image Processing*, New Jersey, U.S.A.: Prentice Hall Information Systems, 1989.

9 P.M. Atkinson. “Optimal ground-based sampling for remote sensing investigations: estimating the regional mean”. *International Journal of Remote Sensing*, 12, 559–567. 1991.

10 S. A. Dudani, K.J. Breeding and R. B. McGhee. “Aircraft Identification by Moment Invariants”. *IEEE Transactions on Computers*, Vol. C-26. No. 1, pp 39-45, 1977.

11 M. Dehghan and K. Faez Farsi. “Handwritten Character Recognition with Moment Invariants*”. International Conference on Digital Signal Processing*, 2:507-510, 1997.

12 G. Granlund. “Fourier Pre-processing for Hand Print Character Recognition”. *IEEE Transactions on Computers*, Vol. C-21, 195-201, 1972.

13 C. Wu. “An improved image retrieval method based on moments invariants”. *Advanced Materials Research*, Volumes 524 – 527.3797, May 2012.

14 A. Kadir, L.E. Nugroho, A. Susanto and P.I. Santosa. “A Comparative Experiment of Several Shape Methods in Recognizing Plants”. *International Journal of Computer Science & Information Technology (IJCSIT),* Vol 3, No 3, June 2011.

15 L. Keyes and A.C. Winstanley. “Using Moment Invariants for Classifying Shapes on Large-scale Maps”. *Computers Environment and Urban Systems*, 25, 119-130, 2001.

16 J. Flusser. “Moment Invariants in Image Analysis”. *Transactions on engineering, computing and technology,* v11 February 2006 issn1305-5313.

17 L. Kotoulas and I. Andreadis. "Image analysis using moments". *Laboratory of Electronics, Section of Electronics and Information Systems Technology, Department of Electrical and Computer Engineering*, Democritus University of Thrace, Greece.

18 J. Leon-Garcia, G. Sanchez-Perez, K. Toscano-Medina, G. Aguilar-Torres, H. Perez-Meana. "Invariant Moments Applied to Fingerprint Recognition". *International Journal of Circuits, Systems and Signal Processing*, Issue 3, Volume 2, 2008.

19 A. Halet, H.T. Huynh, G. Lampropoulos. "La classification des images affectées par le bruit avec la methode des moments invariants homogenes." *3rd Greek conference on image Processing*, Juan Les Pins, France, 1999.

20 G. Dimitoglou, A. J. Adams, M. C. Jim. "Facial Expression Recognition Based on Moment Invariants*". Journal of Computing*, Volume 4, Issue 8, August 2012.

21 M. Husnain, S. Naweed. "English Letter Classification Using Bayesian Decision Theory and Feature Extraction Using Principal Component Analysis". *European Journal of Scientific Research*, ISSN 1450-216X Vol.34 No.2, pp.196-203, 2009.

22 K. Jayech, M. A. Mahjoub. "New approach using Bayesian Network To improve content based image classification systems". *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 6, November 2010.

23 A. Giusto. "A Primer on Learning Vector Quantization", USA: 2005.

24 T. Kohonen. "Self-organizing maps", USA: Springer-Verlag, 2001.

25 S. Thepade, D. Parekh, U. Thapar, V. Tiwari. "LBG Algorithm for Fingerprint Classification*". International Journal of Advances in Engineering & Technology*, Nov. 2012.

26 A. Witoelar, M. Biehl and B. Hammer. "Learning Vector Quantization: generalization ability and dynamics of competing prototypes". *Technical Report, Mathematics and Computing Science,* University Groningen, Groningen, Netherlands: December. 2006.

27 I. Sommer, O. Müller, F. S. Domingues, O. Sander, J. Weickert, T. Lengauer. "Moment invariants as shape recognition technique

. For comparing protein binding sites". *Oxford Journals, Life Sciences, Bioinformatics*, Volume 23, Issue23, Pp. 3139-3146.

28 A. Farag, A. Ali, J. Graham, A. Farag, S. Elshazly and R. Falk. "Evaluation of geometric feature descriptors for detection and classification of lung nodules in low dose CT scans of the chest". *International Symposium on Biomedical Imaging*, (ISBI-11), pp. 169 - 172, March 2011.

29 A. D Patel, T. U. Pandya, C. D. Panchasara, K. N.Mandera. "Identifying Malaria Parasite Using Pattern Recognizing Technique". *World Health Organization. Guidelines for theTreatment of Malaria,* pp. 2056-2150, Geneva: WHO; 2009.

30 Welcome. "Welcome Images (Sickle cell anaemia". AC.UK, EM Unit, UCL Medical School, Royal Free Campus, UK: 2013.

31 H. Jun-Wei, C. Jian-Ming, F. Kao-Chin, H. Chin-Chuan. "Aircraft Type Recognition in Satellite Images". *16th IPPR Conference on Computer Vision*, *Graphics and Image Processing,* CVGIP, 2003.

32 W. Li, S. Xiang, H. Wang, C. Pan. “Robust Airplane Detection in Satellite Images”, *18th International conference on image processing*, 2012.

33 E. Polat, C. Yildiz. “Stationary Aircraft Detection from Satellite Images”. *IU-JEEE*, Vol. 12(2), pp 1523-1528, 2012.

34 K.Roopa, T.V. Ramamurthy. “Aircraft Recognition System Using Image Analysis”. *Emerging Research in Electronics, Computer Science and Technology,* Volume 248, pp 195-204, 2014.

35 F.W. Smith, M.H. Wright. “Automatic Ship Photo Interpretation by the Method of Moments”. *IEEE Transactions on Computers*, Volume: C-20, Issue 9, 1971.

36 R. Singh, A. Kaur “A Comparative Analysis of Speckle Noise Reduction Techniques and Their Effect on Image Edge Localization”. IJCST Vol. 2, Issue 4, Oct. Dec. 2011

37 Y. Mingqiang, K. Kidiyo, R. Joseph. “A Survey of Shape Feature Extraction Techniques”. *IETR-INSA, UMR-CNRS 6164, 35043 Rennes, Shandong University, 250100, Jinan, France, China*. pp 43-90, 2008.

38 J. Kim. “Automatic aircraft recognition and identification”. PhD thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, http://ro.uow.edu.au/theses/499, 2005.

39 R. Gonzalez, R.E. Woods, and S.L. Eddins. *Digital Image Processing Using MATLAB,* 2nd Edition, Upper Saddle River, New Jersey, U.S.A.: Pearson Prentice-
Hall, Pvt. Ltd, 2004.

40 D. Zhang and G. Lu. “Review of shape representation and description techniques”. *Pattern Recognition*, vol. 37, pp. 1-19, 2004.

41 C. Chen, “Improved Moment Invariants for Shape Recognition”. *Pattern Recognition*, Vol. 26, No. 5, pp. 683-686. 1993.

42 S. O. Belkasim, M. Shridhar, and M. Ahmadi. “Pattern Recognition with Moment Invariants: A Comparative Study and New Results”. *Pattern Recognition*, 24: 1117-1138, 1991.

43 P. Dale and M. De Simone, “The Automatic Recognition of features in Digital maps. Land and Minerals Surveying”. Vol. 4, pp. 229-305, 1986.

44 R. B. Yadava, N. K. Nishchala, A. K. Gupta, and V. K. Rastogi. “Retrieval And classification of shape-based objects using Fourier, Generic Fourier, and wavelet-Fourier descriptors technique: A comparative Study”. *Optics and Lasers in Engineering,* vol. 45(6), pp. 695-708, 2007.

45 E. Persoon, and K.S. Fu. “Shape Discrimination Using Fourier Descriptors”. *IEEE Transactions on Pattern analysis and Machine Intelligence*, Vol. PAMI-8, No. 3, pp 388-397. 1986.

46 N. Bouguila, T. Elguebaly. “A Bayesian approach for texture images Classification and retrieval”. *International Conference on Multimedia Computing and Systems*, ICMS.2011.5945719, pp 1-6, Canada, 2011.

47 M. Dixit, N. Rasiwasia, N. Vasconcelos. “Adapted Gaussian models for Image classification”. *IEEE Conference on Computer Vision and Pattern*, CVPR.2011.5995674, pp 937-943, USA, 2011.

48 D. G. Wambaa, E. Mwangi. “Aircrafts identification using moments invariants feature extraction and Bayesian Decision Theory Classification,” SAICSIT 2012 Masters and Doctoral Symposium, Irene Country Lodge Centurion South Africa, 1 October 2012.

49 F. O. Owalla, E. Mwangi. “A Robust Image Watermarking Scheme Invariant to Rotation, Scaling and Translation Attacks”. *16th IEEE Mediterranean Electrotechnical Conference*, March, 2012.

50 F. Achard, F. Blasco. “Analysis of vegetation seasonal evolution and Mapping of forest cover in West Africa with the use of NOAA AVHRR HRPT data”. *Photogrammetric Engineering and Remote Sensing*, 56, 1359–1365, 1990.

51 M. Antonini. “*Transformèe en ondelettes et compression numèrique des Images*”, PhD Thesis, Nice University, France, 1991.

52 J.D. Banfield and A.E. Raftery. “Model-based Gaussian and non Gaussian clustering”. *Biometrics*, 49, 803–829. 1993.

53 L. Ge, S. Xian, F. Kun, W. Honggi. “Aircraft Recognition in High-Resolution Satellite Images Using Coarse-to-Fine Shape Prior”. *Geoscience and Remote Sensing Letters, IEEE* , Volume 10**,** May 1, 2013.

54 N. T. Pham, J.P. Le Cadre. “Vision based aircraft tracking under LPC coordinate system,”*15th IFAC Symposium on System Identification*, Saint-Malo, France: July 6-8. 2009.

55 S. Yan, X. He, Y. Hu, H. Zhang, M. Li, and Q. Cheng. “Bayesian shape localization for face recognition using global and local textures”. IEEE transactions on circuits and systems for video technology vol 14 Jan 2004.

56 Y. Zhou, L. Gu, H. Zhang. “Bayesian tangent shape model: estimating shape and pose parameters via Bayesian inference”. IEEE international conference on computer vision and pattern recognition, Wisconsin, USA, June 2003.

57 “Aircraft Images”. http://fas.org/programs/ssp/man/miltutorials/ aircraft.html

58 “ Military Aircrafts”. http://www.avion-de- /Combat_thum/guerre-avion.jpg

59 “Flight Global”. http//www.flightglobal.com/insight

# APPENDIX

This appendix is divided into two parts. Part A1 contains a paper that was published during this research [48] and information about the conference. Part A2 presents the MATLAB code simulations that were used to generate the results presented in chapter 5 of this thesis.

# APPENDIX A

# MATLAB CODE

## A1 Boundary Signature

```
% Import an image from any supported graphics image file format, in any % of the supported bit depths,
% this example reads a truecolour image into the MATLAB workspace as % the variable RGB.
b = imread ('C: \Users\wambaa\Desktop\AIRCRAFTS DATABASE\X-45-C.jpg');
% B =bwboundaries (b) traces the exterior boundaries of objects, % eliminating boundaries of holes inside these objects, in the binary image % b.
[B, L] = bwboundaries (b,'noholes');
% Transform Cartesian coordinates to polar or cylindrical
for k = 1: length (B)
boundary = B {k};
[theta, rho] = cart2pol ((boundary (: 2)-280),-(boundary (: 1)-280));
% Convert angles to degrees.
plot (theta*180/pi,rho,'k','LineWidth',0.5)
end
```

## A2 Image Enhancement

```
% Read and show the image.
I = imread('C:\Users\wambaa\Desktop\AIRCRAFTS DATABASE\X-45-C.jpg');
imshow(I);
% morphologically open image.
% imopen performs morphological opening on the grayscale or binary
% image with a structuring element
% the argument must be a single structuring element object, as opposed % to an array of objects.
% Morphological open operation is an erosion followed by a dilation, using % the same structuring element
% for both operations.
background = imopen (I, strel('disk',30));
% Display the Background Approximation as a Surface figure, surf (double (background (1:8: end, 1:8: end))), zlim ([0 255]);
set (gca,'ydir','reverse');
I2 = I - background;
figure, imshow (I2);
I3 = imadjust (I2);
figure, imshow (I3);
level = graythresh (I3);
bw = im2bw (I3, level);
bw = bwareaopen (bw, 50);
imshow (bw);
title ('Binary Image')
```

## A3 Edge Enhancement

```
% creates a predefined 2-D filter h of the specified type.
```

```
B=imread ('C: \Users\wambaa\Desktop\AIRCRAFTS DATABASE\X-45-C.jpg')
G=fspecial ('unsharp', 0.5);
% filters the data with a two-dimensional FIR filter. It computes the result, using % two-dimensional
% correlation, and returns the central part of the correlation that is the same size % as X.
BG = filter2 (G, B)
Imshow (B), figure, imshow (BG/255)
```

### A4 Gray Threshold

```
B = imread ('C: \Users\wambaa\Desktop\AIRCRAFTS DATABASE\X-45-C.jpg');
imshow (B);
% Create a binary version of the image so you can use toolbox functions.
threshold = graythresh (B, []);
figure, imshow (threshold);
```

### A5 Median Adaptive Filter

```
% 2-D median filtering
B=imread ('C: \Users\wambaa\Desktop\AIRCRAFTS DATABASE\X-45-C.jpg');
L = medfilt2 (B, [3 3]);
figure, imshow (L)
```

### A6 Lee Adaptive filter

```
function outputImage=fcnFirstOrderStatisticsFilter (inputImage, mask)
%fcnFirstOrderStatisticsFilter performs noise filtering on an image based
% on using First Order Local Statistics around a prespecified pixel
% neighbourhood.
InputImage=imread('C:\Users\Wambaa\Desktop\AIRCRAFTS
DATABASE\cameraman001.tif');
imshow (inputImage);
ImageType=class (inputImage);
mask = getnhood (strel ('square', 5));
windowSize=size(mask);
InputImage=padarray(inputImage,[floor(windowSize(1)/2),floor(windowSize(2)/2)],'symmetri
c','both');
inputImage=double((inputImage));
[nRows,nCols]=size (inputImage);
localMean=zeros([nRows nCols]);
localVar=zeros([nRows nCols]);
k=zeros(nRows, nCols);
for i=ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2)
   for j=ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2)
      localNeighborhood=inputImage(i-floor(windowSize(1)/2):i+floor(windowSize(1)/2),...
         j-floor(windowSize(2)/2):j+floor(windowSize(2)/2));
      localNeighborhood =localNeighborhood(mask);
      localMean(i,j)= mean(localNeighborhood(:));
      localVar(i,j) = var(localNeighborhood(:));
   end
end
localNoiseVar=localVar(ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2))./(localMean(ceil(windowSize(1)/2):...
```

```
nRows-floor(windowSize(1)/2),ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2))+eps);
globalNoiseVar = sum(localNoiseVar(:));
k(ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2),ceil(windowSize(2)/2):nCols-
floor(windowSize(2)/2))...
=(1-((localMean(ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-
floor(windowSize(2)/2)).^2).*(localVar(ceil(windowSize(1)/2):nRows-
floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-
floor(windowSize(2)/2)))./(localVar(ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2))*(1+globalNoiseVar)));
outputImage = (localMean(ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2))+(k(ceil(windowSize(1)/2):nRows-
floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-
floor(windowSize(2)/2)).*(inputImage(ceil(windowSize(1)/2):nRows-
floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2))-
localMean(ceil(windowSize(1)/2):nRows-floor(windowSize(1)/2),...
ceil(windowSize(2)/2):nCols-floor(windowSize(2)/2)))));
outputImage = cast(outputImage,imageType);
imshow(outputImage)
```

**A7 MSE/ PSNR**

```
% PSNR of a grayscale image. The PSNR block computes the peak signal-to-noise ratio,
% in decibels, between two images. This ratio is often used as a quality measurement
% between the original and a filtered image. The higher the PSNR, the better the quality
% of the filtered, or reconstructed image.
% Clean up.
clc;
close all;
clear;
workspace;
format long g;
format compact;
fontSize=20;
% Get Images
grayImage=imread('C:\Users\Wambaa\Desktop\Phone                    Memory
Copy\DSP\Images\standard_test_images\cameraman.tif');
[rows columns]=size(grayImage);
% Display the first image
figure;imshow(grayImage,[]);
title('Filtered Gray scale Image','FontSize',fontSize);
set(gcf,'Position',get(0,'Screensize'));% Maximum figure.
% Get a second image by adding noise to the first image.
noisyImage=imnoise(grayImage,'speckle',0.03);
% Display the second image
figure;imshow(noisyImage,[]);
title('Noisy Gray Scale Image','FontSize',fontSize);
% PSNR Calculation
% First The Square Error Image.
```

```
squaredErrorImage = (double(noisyImage)-double(grayImage)).^2;
% Mean Squared Error.
mse = sum(sum(squaredErrorImage))/(rows*columns);
PSNR = 10*log10(256^2/mse);
message = sprintf(' The mean square error is %.2f.\n The PSNR = %.2f',mse,PSNR)
msgbox(message)
```

## A8 Features Extraction

### A8.1. Invariant Moments

```
function phi = invmoments(F)
% Author: Wambaa D.G.
% invmoments computes invariant moments of an image.
% phi= invmoments computes the moment invariants of the image
% F.phi is a seven element row vector containing
% the moment invariants
% F must be a 2-D,real ,nonsparse,numeric or logic matrix.
F=imread('C:\Users\wambaa\Desktop\AIRCRAFTS DATABASE\X-45-C.jpg');
if (ndims(F) ~= 2) || issparse(F) || ~isreal(F) || ~(isnumeric(F) || ...
 islogical(F))
error(['F must be a 2-D,real,non-sparse,numeric or logical '...
'matrix.']);
end
F = double(F);
phi = compute_phi(compute_eta(compute_m(F)));
%.....................................................................%
function m = compute_m(F)
[M,N] = size(F);
[x,y] = meshgrid(1:N,1:M);
% Turn x,y, and F into column vectors to make the summation a bit
% easier to compute in the folowing.
x = x(:);
y = y(:);
F = F(:);
m.m00 = sum(F);
% Protect against divide by zero warnings.
if (m.m00 == 0)
    m.m00 = eps;
end
% The other central moments:
m.m10 = sum(x.*F);
m.m01 = sum(y.*F);
m.m11 = sum(x.*y.*F);
m.m20 = sum(x.^2.*F);
m.m02 = sum(y.^2.*F);
m.m30 = sum(x.^3.*F);
m.m03 = sum(y.^3.*F);
m.m12 = sum(x.*y.^2.*F);
m.m21 = sum(x.^2.*y.*F);
%.....................................%
function e = compute_eta(m)
xbar = m.m10/m.m00;
ybar = m.m01/m.m00;
```

```
e.eta11 = (m.m11-ybar*m.m10)/m.m00^2;
e.eta20 = (m.m20-xbar*m.m10)/m.m00^2;
e.eta02 = (m.m02-ybar*m.m01)/m.m00^2;
e.eta30 = (m.m30-3*xbar*m.m20+2*xbar^2*m.m10)/m.m00^2.5;
e.eta03 = (m.m03-3*ybar*m.m02+2*ybar^2*m.m01)/m.m00^2.5;
e.eta21 = (m.m21-2*xbar*m.m11-ybar*m.m20+2*xbar^2*m.m10)/m.m00^2.5;
e.eta12 = (m.m12-2*ybar*m.m11-xbar*m.m02+2*ybar^2*m.m10)/m.m00^2.5;
%.....................................................................%
function phi = compute_phi(e)
phi(1)= abs(log(e.eta20+e.eta02));
phi(2)=abs(log((e.eta20-e.eta02)^2+4*e.eta11^2));
phi(3)=abs(log((e.eta30-3*e.eta12)^2+(3*e.eta21-e.eta03)^2));
phi(4)=abs(log((e.eta30+3*e.eta12)^2+(e.eta21+e.eta03)^2));
phi(5)=abs(log((e.eta30-3*e.eta12)*(e.eta30+e.eta12)*...
 ((e.eta30+e.eta12)^2-3*(e.eta21+e.eta03)^2)+...
 (3*e.eta21-e.eta03)*(e.eta21+e.eta03)*...
 (3*(e.eta30+e.eta12)^2-(e.eta21+e.eta03)^2)));
phi(6)=abs(log((e.eta20-e.eta02)*((e.eta30+e.eta12)^2-...
 (e.eta21+e.eta03)^2)+...
  4*e.eta11*(e.eta30+e.eta12)*(e.eta21+e.eta03)));
phi(7)=abs(log((3*e.eta21-e.eta03)*(e.eta30+e.eta12)*...
 ((e.eta30+e.eta12)^2-3*(e.eta21+e.eta03)^2)+...
 (3*e.eta12-e.eta30)*(e.eta21+e.eta03)*...
 (3*(e.eta30+e.eta12)^2-(e.eta21+e.eta03)^2))).';
```

*%Note that the absolute value of the log was used instead of the moment invariant %values themselves. This is so as to reduce the dynamic range and the absolute %values avoids having to deal with the complex numbers that result when %computing log of negative moments invariants because interest lies in the %invariance of the moments and not the sign.*

**A9 Euclidean Distance**

**function d = DistMatrix(A,B)**

*% Return distance matrix between point A($x_1, x_2, x_3, x_4, x_5, x_6, x_7$) and*
*% B($y_1, y_2, y_3, y_4, y_5, y_6, y_7$)*
*% Author: Wambaa D.G,.*

```
[hA,wA] = size(A);
[hB,wB] = size(B);
if hA == 1 & hB == 1
d=sqrt(dot((A-B),(A-B)));
else
C= [ones(1,hB);zeros(1,hB)];
D=flipud(C);
E= [ones(1,hB);zeros(1,hB)];
F=flipud(C);
G= [ones(1,hB);zeros(1,hB)];
H=flipud(C);
I= [ones(1,hB);zeros(1,hB)];
J= [ones(1,hA);zeros(1,hA)];
K=flipud(J);
L= [ones(1,hA);zeros(1,hA)];
M=flipud(L);
N= [ones(1,hA);zeros(1,hA)];
O=flipud(N);
P= [ones(1,hA);zeros(1,hA)];
```

```
Q=A*C; R=A*D; S=A*E; T=A*F;   U=A*G; V=A*H; W=A*I; X=B*J; Y=B*K; Z=B*L; AA=B*M;
AB=B*N; AC=B*O; AD=B*P;
d=sqrt((Q-X').^2+(R-Y').^2+(S-Z').^2+(T-AA').^2+(U-AB').^2+(V-C').^2…...
+(W-AD').^2
end
```

## A10 Classification

### A10.1. Covariance Matrix

```
function  [C, M] = covmatrix(X)
% COVMATRIX  Computes the covariance matrix of a vector population.
% [C,M] = COVMATRIX (X) computes the covariance matrix C and the mean
% vector M of a
% vector population organized as the rows of matrix X. C is of size N by N and M % is of size
N by
% 1 , where N is the dimension of the vectors  ( the number of columns of X).
[K,n] = size(X);
X = double(X);
If n==1 % Handle special case
C = 0;
M = X;
else
% Compare an unbiased estimate of M.
M = sum(X,1)/K;
% Substract the mean from each row of X.
X = X-M(ones (K,1), :;
% Compute an unbiased estimate of C. Note that the product is
%  X'*X because the vectors are rows of X.
C = (X'*X)/(K-1);
M = M'; % Convert b to a column vector.
end
```

### A10.2. Bayesian Classification

```
function d = bayesgauss(X, CA, MA P)

% BAYESGAUSS Bayes Classifier for Gaussian Patterns.
% D= BAYESGAUSS(X, CA, MA, P) computes the Bayes decision
% functions of the patterns in the rows of array X using the covariance
% matrices   and mean vectors provided in the arrays CA and MA.
% CA is an array of n-by-n-by-W, where n is the dimensionality of the
% patterns and W is the number of classes. Array MA is of dimension
% n-by-W i.e the columns of MA are the individual mean vectors.
% The location of the covariance matrices and the mean vectors in their
% respective arrays must correspond. %There must be a covariance
% matrix and a mean vector for each pattern class, even if some of the %covariance matrices
and or mean vectors are equal. X is an array of
% k-by-n, where k is the total number of patterns to be classified i.e the
% pattern vectors are rows of X. P is a 1-by-W  array , containing
```

```
% the probabilities of occurrence of each class. If P is not included in the % argument list,
the classes are assumed to be equally likely.The output % D is a column vector of length K.
Its I^th element is the class number
% assigned to the I^th vector in X during Bayes classification.
d = [ ]; % initialize d.
error(nargchk(3,4,nargin)) % verify correct number of inputs.
n = size(CA, 1); % Dimensions of patterns
% Protect against the possibility that the class number is included as an
% (n+1)^th element of the vectors.
X = double(X(:, 1:n));
W = size(CA,3);% Number of pattern classes.
K = size(X,1); % Number of patterns to classify.
if nargin ==3
P(1:W) = 1/W; %Classes assumed equally likely.
else
if sum(P) ~= 1
error(' Elements of P must sum to 1.');
end
end
% Compute the determinants.
for J = 1:W

DM(J) = det(CA(:, :, J));

end
% Compute inverses , using right division (IM/CA), where IM =
% eye(size(CA, 1 )) % is the n-by-n % identity matrix .Reuse CA to
% conserve memory.
IM = eye(size(CA,1));
for J = 1:W
CA(:, :,J) = IM/CA(;, :, J);
end
% Evaluate the decision functions. The sum terms are the Mahalanobis
% distances.
MA = MA' ; % Organize the mean vectors as rows.
for I = 1:K

for J = 1:W
m = MA(J, ;);
Y = X-m(ones(size(X,1),1), :) ;
if P(J) == 0
D(I, J) = -inf;
else
D(I, J) = log(P(J)) – 0.5*log(DM(J)) …
-0.5*sum(Y(I, :) * (CA(:, :, J) * (I, :) '));
end
end
end
% Find the maximum in each row of D. These maxima give the class of
% each pattern :
for I = 1:K
J = find(D(I, :) == max(D(I, : )));
D(I, :) = J(:);
End
```

*% when there are multiple maxima the decision is arbitrary .Pick the first % one.*

D = d(:, 1);

## APPENDIX B

**B.1 Vector Quantization** *k*-means uses a two-phase iterative algorithm to minimize the sum of point-to-centroid distances, summed over all *k* clusters:

1. The first phase uses batch updates, where each iteration consists of reassigning points to their nearest cluster centroid, all at once, followed by recalculation of cluster centroids. This phase occasionally does not converge to solution that is a local minimum, that is, a partition of the data where moving any single point to a different cluster increases the total sum of distances. This is more likely for small data sets. The batch phase is fast, but potentially only approximates a solution as a starting point for the second phase.
2. The second phase uses online updates, where points are individually reassigned if doing so will reduce the sum of distances, and cluster centroids are recomputed after each reassignment. Each iteration during the second phase consists of one pass though all the points. The second phase will converge to a local minimum, although there may be other local minima with lower total sum of distances. The problem of finding the global minimum can only be solved in general by an exhaustive (or clever, or lucky) choice of starting points, but using several replicates with random starting points typically results in a solution that is a global minimum.
3. `IDX = kmeans(X,k)` partitions the points in the `n`-by-`p` data matrix `X` into `k` clusters. This iterative partitioning minimizes the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. Rows of *X* correspond to points, columns correspond to variables. *k*`-means` returns an `n`-by-`1` vector `IDX` containing the cluster indices of each point. By default, `kmeans` uses squared Euclidean distances.
4. `[IDX,C] = kmeans(X,k)` returns the `k` cluster centroid locations in the `k`-by-`p` matrix `C`.
5. `[IDX,C,sumd] = kmeans(X,k)` returns the within-cluster sums of point-to-centroid distances in the `1`-by-`k` vector `sumd`.
6. `[IDX,C,sumd,D] = kmeans(X,k)` returns distances from each point to every centroid in the `n`-by-`k` matrix `D`.
7. `[...] =` *k*`-means(...,param1,val1,param2,val2,...)` enables you to specify optional parameter/value pairs to control the iterative algorithm used by *k*`-means`.

**B1.2. Unsupervised Classification function y = kMeansCluster(m,k,isRand)**

```
% kMeansCluster
% Author:
% Purpose: Classify the objects in data matrix based on the attributes.
% Criteria: Minimize Euclidean distance between centroids and object points.
% Output: matrix data plus an additional column for the class of each object.
% k = 5 : number of classes
% Input: m, matrix data:objects in rows and attributes in columns.
% Local variables
% f :row number of data that belong to class i
% c : centroid coordinate size(1:k, 1:maxCol)
% g : current iteration class matrix size (1:maxRow)
% I : scalar iterator
% maxCol : scalar number of rows in the data matrix m = number of attributes
% maxRow :scalar number of columns in the data matrix m= number of objects
% termp : previous iteration group matrix size (1:maxRow)
if nargin<3, isRand = 0; end
if nargin <2, k=1; end
[maxRow,maxCol] = size(m)
if  maxRow<=k,
y = [m,1:maxRow]
else
% initial value of centroid
if  isRand,
p = randperm(size(m,1)); % random initialization
for i=1 : k
c(i:) = m((p(i),:)
end
else
for i=1:k
c(I,:) = m(I,:)   % Sequential initialization
end
end
temp = zeros(maxRow,1); % initialize as zero vector
while 1,
d = DistMatrix(m,c); % Calculate objects centroid distances
[g] = min(d,[],2); % find class matrix g
if g == temp,
break;  % stop the iteration
else
temp = g;        % copy class matrix to temporary variable
end
for i = 1:k
f = find(g == i);
 if f      % only compute centroid if f is not empty
c(I,:) = mean(m(find(g == i), :) ,1);
end
end
end
y = [m,g];
end
```

**function e = separation(m,k,isRand)**

```
size(X);
idx10 = kmeans(X,10,'distance','sqEuclidean','display','iter');
```

```
[silh10,h] = silhouette(X,idx10,'sqEuclidean');
set(get(gca,'Children'),'FaceColor',[.8 .8 1])
xlabel('Silhouette Value')
ylabel('Cluster')
mean(silh10)
```


SOUTH AFRICAN INSTITUTE FOR COMPUTER SCIENTISTS AND INFORMATION TECHNOLOGISTS

**Masters and Doctoral Symposium**

1 October 2012

Irene Country Lodge, Centurion

# APPENDIX C

# Aircrafts identification using moments invariants feature extraction and Bayesian Decision theory classification

Dickson G. Wambaa
Student Member IEEE
Electrical and Information Dept.
University Of Nairobi, Kenya

Elijah Mwangi
Member IEEE
Electrical and Information Dept.
University Of Nairobi, Kenya

## ABSTRACT

In *the* present study the method of moments invariants has been applied to produce a set of normalized invariant moments to identify aircrafts from a database of aircrafts. Invariance to translation, scaling and rotation allows considerable robustness when applied to images of aircrafts**.** Bayesian Decision Theory is applied in the classification of the images.

## Keywords

Computer vision, Moment invariants, Bayesian Theory, Classification, Features, Lee filter.

## INTRODUCTION

The ability to reliably identify aircraft is an important aspect of air traffic safety. Civilian air traffic controllers need to be constantly updated on the status of aircraft moving through the local airspace. In military scenarios, the need to reliably identify aircraft is even more stringent, since erroneous identification could easily result in friendly fire incidents. In the present study Aircraft Satellite Images are Identified Using Bayesian Decision Theory Classification and Moment Invariants Feature Extraction. The Bayesian Classification is based on the statistical properties of a training set: a collection of

Measurements for which the corresponding class is known. The resulting statistical classifier assigns a measurement to the class which most likely generated the measurement. Also each training example increases the probability that the identification is correct and serves as a benchmark for other classifications. Invariance to translation, scaling and rotation allows considerable robustness when applied to satellite images of aircrafts. The satellite images are normally contaminated by speckle noise and Lee filter has been applied in the image enhancement stage. The study is demonstrated by numerical matlab simulation performed using 2D aircraft images.

## MOMENT INVARIANTS

The 2-D moment of order (p+q) of a digital image f(x, y) is defined as

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad (2.1)$$

for p,q = 0,1,2…., where the summations are over the values of the spatial coordinates x and y spanning the image.

The corresponding central moment is defined as



$$\mu_{pq} = \sum_x \sum_y (x-x_c)^p (y-y_c)^q f(x, y) \quad (2.2)$$

Where

$$x_c = m_{10}/m_{00} \text{ and } y_c = m_{01}/ m_{00} \quad (2.3)$$

Normalized central moment of order (p+q) is defined as

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma} \quad (2.4)$$

for p,q = 0,1,2,….. Where

$$\gamma = (p+q)/2 + 1 \quad (2.5)$$

For p+q = 2,3….

Individual moments values do not have the descriptive power to uniquely represent arbitrary shapes nor possess the required invariance characteristics, but sets of functions based on these moments can be determined that do so. A set of seven 2-D moment invariants that are insensitive to translation, scaling, rotation and reflection can be derived from the above equations.

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2$$

$$\phi_5 = (3\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \times [3(\eta_{30} + \eta_{12})_2 - (\eta_{21} + \eta_{30})^2]$$

(2.6)

## 3 BAYESIAN CLASSIFICATION

It is a fundamental statistical approach to the generic pattern classification problems. It makes the assumption that the solution to pattern classification problem is purely based on probabilistic values and all the relevant probability values are known. The decision rule of Bayesian Decision Theory says that for minimum error rate classifier, we should choose the class with minimum posterior probability.

The explanation is as under:

let **λ** be finite set of classes ***c1, c2, c3 … cn*** and our unknown feature vector ***x***, where ***x*** (∈ R) is a *d*-dimensional vector. After calculating conditional posterior probabilities of every class of **λ**, choose the class **c**∈**λ** for which the posteriori is maximum. The estimation of $P(Ci \mid x)$ depends on estimated value of $P(x \mid Ci)$, the likelihood. The generic Bayes Rule is given by

$$P(C_n \mid x) = \frac{P(x \mid C_n)P(C_n)}{P(x)} \quad (3.1)$$

Where $P(C)$ is prior probability and $P(x)$ is marginal probability, aka evidence. Where

$P(x)$ is calculated by

$$P(x) = \sum_{n=1}^{m} p(x/C_n)P(c_n)$$

Eq (3.1) can be written as

$$posterior = \frac{likelihood\ X\ prior}{evidence}$$

The main problem in pattern classification problems is to calculate the conditional probability density values $P(x|C_n)$ which are unknown. As feature vector $\boldsymbol{x}$ is generated by a per-class prototype $\boldsymbol{x_i}$ where the dataset is in bulk, the distribution of the data is assumed to be Normal (Gaussian). This assumption plays a vital role in correct prediction of pattern classification. The value $P(x \mid C)$ can be predicted by other techniques like logistic regression but this is not the objective of this paper. Our target so far is to find the class with maximum posterior probability max $\lambda$ $P(C \mid x)$ with minimum error rate, not just $P(x \mid C)$. The statistical decision theory can be formulated resorting to the Bayes theory introducing the concept of a risk defined as the expected value of the error cost function. If the latter is assumed to be either a quadratic function or a uniform function, then the Maximum A posteriori Probability (MAP) inference solutions can be calculated.

## 4. PREPROCESSING

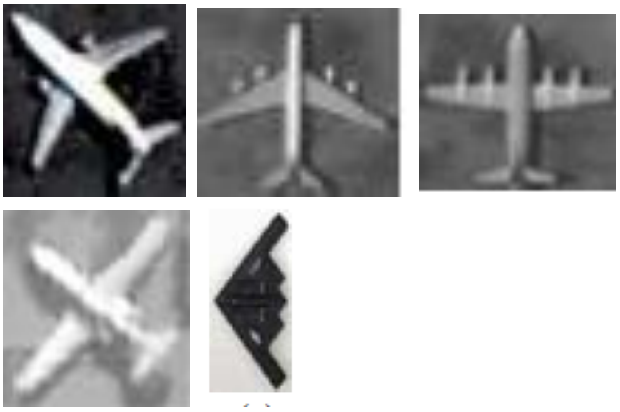


Fig 4.1 Contaminated satellite aircraft images.

Images are contaminated by noise through

i. Imperfect instruments
ii. Problems with data acquisition PROCESS
iii. Natural phenomena interference
iv. Transmission errors.

The type of noise found in satellite images is speckle noise and this determines the algorithm used in denoising. Fig 4.1

This is a multiplicative noise. The distribution noise can be expressed by:

$$J = I + n*I$$

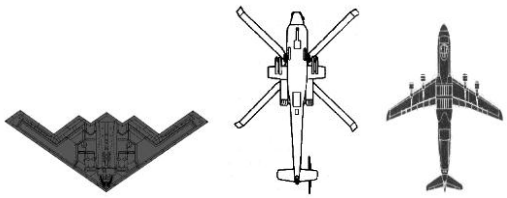
- Where, *J is the distribution speckle noise image, I is the input image and n is the uniform noise image.*

The mathematical function determines the filter type.

i. Mean filter-averages the window pixels.
ii. Median filter-calculates the median pixel.
iii. Lee-sigma and Lee filters-use statistical distribution of pixels in the window.
iv. Local region filter-compares the variances of window regions.
v. The Frost filter replaces the pixel of interest with a weighted sum of the values within the nxn moving window and assumes a multiplicative noise and stationary noise statistics.

- Lee filter is applied so as to reduce the multiplicative noise to an additive noise that is easier to handle.

## 5 Results



B2 Bomber AH64 Helicopter C5 Carrier

Fig 5.1 The Training set Classes C1, C2, and C3 respectively

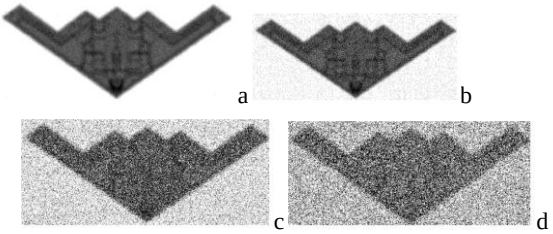


Fig 5.2 an aircraft test image with noise contamination variance of 0.01, 0.02, 0.03, and 0.04 respectively.

Fig 5.3 Binarized and Thresholded image.

Table 5.1 Extracted Geometrical feature vectors of test set

| Classes | $Ø_1$ | $Ø_2$ | $Ø_3$ | $Ø_4$ | $Ø_5$ | $Ø_6$ | $Ø_7$ |
|---|---|---|---|---|---|---|---|
| B2 Class 1 | 6.6132 | 14.0538 | 15.2462 | 17.4521 | 33.9469 | 24.6798 | 39.2648 |
| AH 64 Class 2 | 7.1729 | 16.6723 | 19.7413 | 21.8784 | 42.8038 | 30.2146 | 47.1336 |
| C5 Class 3 | 7.1487 | 20.2793 | 22.4129 | 24.4962 | 48.0614 | 34.6401 | 50.1980 |

| | $Ø_1$ | $Ø_2$ | $Ø_3$ | $Ø_4$ | $Ø_5$ | $Ø_6$ | $Ø_7$ |
|---|---|---|---|---|---|---|---|
| **B2 ORIGINAL** | **6.6132** | **14.0538** | **15.2462** | **17.4521** | **33.9469** | **24.6798** | **39.2648** |
| **B2 NOISE FILTERED** | 6.6001 | 13.9810 | 15.1678 | 17.4434 | 33.8456 | 24.6578 | 40.9765 |
| **B2 ( 0.01 VAR)** | 6.5579 | 13.9115 | 15.0382 | 17.2442 | 33.5329 | 24.4031 | 41.0169 |
| **B2 ( 0.02 VAR)** | 6.5406 | 13.8898 | 14.9673 | 17.1737 | 33.3923 | 24.3223 | 38.6145 |
| **B2 ( 0.03 VAR)** | 6.4703 | 13.7136 | 14.6351 | 16.8403 | 32.7292 | 23.9045 | 38.4642 |
| **B2 ( 0.04VAR)** | 6.4124 | 13.5763 | 14.2593 | 16.4614 | 31.9765 | 23.4609 | 36.7216 |

Table 5.2 Extracted Geometrical feature vectors of added noise images.

## 6 Bayesian Decision

For a minimum error rate classifier the choice is on the class with maximum posterior probability.

Let λ be set of 3 classes $C_1$, $C_2$, $C_3$. x be an unknown feature vector of dimension 7.Calculate the conditional posterior probabilities of every class $C_i$ and choose the class with maximum posteriori probability. 3 classes of Data which are all likely to happen therefore

$$P(C_i) = 0.333$$

$$\text{Posterior} = \frac{\text{likelihood x prior}}{\text{Evidence}}$$

$$P(C_i \backslash x) = \frac{P(x \backslash C_i)\, P(C_i)}{P(x)}$$

$$\text{Posterior (AH 64)} = \frac{P(\text{AH 64})\, P(x/\text{ AH 64})}{P(\text{evidence})}$$

$$\text{Posterior (C5)} = \frac{P(\text{C5})\, P(x/\text{ C5})}{P(\text{evidence})}$$

$$\text{Posterior (B2)} = \frac{P(\text{B2})\, P(x/\text{ B2})}{P(\text{evidence})}$$

| **Posterior probability** | **FILTERED SAMPLE** | **SAMPLE 1** | **SAMPLE 2** | **SAMPLE 3** | **SAMPLE 4** | **SAMPLE 5** |
|---|---|---|---|---|---|---|
| AH 64 | $1.6954X10^{-2}$ | $\mathbf{1.6789X10^{-2}}$ | $\mathbf{1.6034X10^{-2}}$ | $\mathbf{1.5674X10^{-2}}$ | $\mathbf{1.5045X10^{-2}}$ | $\mathbf{1.4567X10^{-2}}$ |
| C5 | $1.9653X10^{-2}$ | $1.8965X10^{-2}$ | $1.8463X10^{-2}$ | $1.8062X10^{-2}$ | $1.7453X10^{-2}$ | $1.6666X10^{-2}$ |
| B2 | $2.4239X10^{-2}$ | $2.2346X10^{-2}$ | $2.21567X10^{-2}$ | $2.1866X10^{-2}$ | $1.9889X10^{-2}$ | $1.8976X10^{-2}$ |

Table 6.1 Posterior Probability

## 7 CONCLUSION

While combining moment features extraction with Bayesian classification and using Lee filters in preprocessing decreases the error rate of identification as compared to non-use of the filters or use of other types of filters this is seen by the increase of the posterior probability values.

## 8 REFERENCES


Richard O. Duda,Peter E. Hart and David G. Stork.Pattern Classification 2nd edition John Wiley and Sons,US,2007

Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins. Digital image processing using matlab 2nd edition Pearson/Prentice Hall,US,2004

William K. Pratt. Digital image processing 4th edition John Wiley,US,2007

Anil K. Jain. Fundamentals of Digital Image Processing Prentice Hall,US,1989

Wei Cao, Shaoliang Meng, "Imaging systems and Techniques",IEEE International Workshop, IST.2009.5071625,pp 164-167, Shenzhen, 2009

Bouguila.N, Elguebaly.T , "A Bayesian approach for texture images classification and retrieval",International Conference on Multimedia Computing and Systems, ICMS.2011.5945719,pp 1-6,Canada, 2011

Mukesh C. Motwani,Mukesh C. Gadiya,Rakhi C. Motwani, Frederick C. Harris Jr., "Survey of Image Denoising Techniques"

, University of Nevada Reno, US, 2001

Conference on Computer Vision and Pattern,

CVPR.2011.5995674, pp 937-943, USA, 2011