FACE RECOGNITION USING SCALE INVARIANT FEATURE TRANSFORM AND BACK PROPAGATION NEURAL NETWORK

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
MOHAMED-A-BASHER ASAGHER

In Partial Fulfillment of the Requirements for the Degree of Master of Science in

Electrical and Electronics Engineering

NICOSIA, 2016

MOHAMED-A-BASHER ASAGHER

FACE RECOGNITION USING SCALE INVARIANT FEATURE TRANSFORM AND BACK PROPAGATION NEURAL NETWORK

NEU 2015

FACE RECOGNITION USING SCALE INVARIANT FEATURE TRANSFORM AND BACK PROPAGATION NEURAL NETWORK

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
MOHAMED-A-BASHER ASAGHER

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronics Engineering

NICOSIA, 2016

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name:

Signature:

ACKNOWLEDGMENTS

I truly feel very thankful to my supervisor Assist. Prof. Dr. Kamil Dimililer for his assistance, guidance and supervision of my thesis. I appreciate his continuous follow up, support and motivation. He was always sharing his time and effort whenever I need him.

I also appreciate NEU Grand Library administration members for offering perfect environment for study, research and their efforts to provide the updated research materials and resources.

I also send my special thanks to my mother for her care, prayers and her passion. I also appreciate my father's continuous support, advice and encouragement. I would also like to say thanks to my wife for her attention, support and availability when I need her.

Finally, I also have to thank God for everything and for supplying me with patience and supporting me with faith.

ABSTRACT

The thesis aims to develop a face recognition intelligent system based on Scale Invariant Feature Transforms “SIFT” algorithm for the feature extraction and backpropagation neural network for classification. The purpose of this research is to evaluate the effectiveness of a backpropagation neural network in recognizing different faces based on SIFT as feature extractor of an average of 128 features and to compare the obtained results with those in the literature review. The developed framework consists of two main phases which are the processing phase and the classification phase in which the image is classified as different faces. In the image processing phase the face images are pre-processed using many techniques such as conversion to grayscale and filtering using median filter. Then the most significant technique takes place which is the feature extraction using SIFT. These techniques are done in order to enhance the quality of images and to extract the important features in such a way to take only the important face’s features and ignoring the other features and parts of the image. At the end of this phase, the images are fed to a backpropagation neural network in which they are classified as different faces for different individuals.

Experimentally, the proposed intelligent face recognition system outperforms many related previous researches as accuracy rate. However, the system explores a bit long processing time. This is due to the use of SIFT algorithm which generally takes long time to perform its 4 steps that lead to the extraction of the features of the face’s image.

Keywords: backpropagation; face recognition; feature extraction; intelligent system; neural network***;*** scale invariant feature transforms; sift

ÖZET

Tez Ölçeği Değişmeyen Feature sınıflandırma özellik çıkarımı ve geri iletme sinir ağı algoritması "SIFT" Dönüştürüyor dayalı bir yüz tanıma akıllı bir sistem geliştirmeyi amaçlamaktadır. Bu araştırmanın amacı, 128 özelliklerinin ortalama özelliği çıkarıcı olarak SIFT dayalı farklı yüzleri tanımada bir geri yayılım sinir ağının etkinliğini değerlendirmek ve literatür olanlarla elde edilen sonuçları karşılaştırmaktır. Geliştirilen çerçeve işleme aşaması ve görüntü farklı yüzleri olarak sınıflandırılan olduğu sınıflandırma aşaması iki ana aşamadan oluşmaktadır. görüntü işleme safhasında yüz görüntüleri önceden işlenmiş gibi gri dönüşüm gibi birçok teknikler kullanılarak ve medyan filtre kullanarak filtreleme vardır. Sonra en önemli teknik elemek kullanarak özellik çıkarma olduğunu gerçekleşir. Bu teknikler görüntü kalitesini artırmak ve sadece önemli yüzün özelliklerini almak için böyle bir şekilde önemli özelliklere ayıklamak ve görüntünün diğer özellikleri ve parçaları göz ardı etmek için yapılır. Bu aşamanın sonunda, görsel farklı bireyler için farklı yüzleri olarak sınıflandırılır bir geri yayılım nöral şebekesine beslenir. Deneysel, önerilen akıllı yüz tanıma sistemi doğruluk oranı gibi birçok ilgili önceki araştırmalar geride bırakıyor. Ancak, sistem biraz uzun işlem süresini araştırıyor. Bu genellikle yüzünün görüntü özelliklerinin çıkartılmasına neden olan 4 adımları gerçekleştirmek için uzun zaman alır SIFT algoritması kullanımı nedeniyle.

Anahtar Kelimeler: akıllı sistem; backpropagation; ölçek değişmeyen özelliği dönüşümleri; özellik çıkarma; sift; sinir ağı; yüz tanıma

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Face Recognition Technology (FRT) is an exploration area traversing a few disciplines, for example, image processing, pattern recognition, computer vision and neural systems. There are numerous utilizations of FRT. These applications range from coordinating of images to continuous coordinating of observation videos. Contingent upon the specific application, FRT has diverse level of trouble and requires extensive variety of strategies and techniques. In 1995, a survey paper by (Chellappa et al., 1995) gives a thorough study of FRT around then. During the previous couple of years, FRT is still under fast development.

Face recognition by people is a characteristic procedure that we perform on everyday life. A brisk look at a face then we can perceive the face and, more often than not, name the individual. Such a procedure happens so rapidly that we never consider what precisely we took a feature at in that face. A few of us may take a more drawn out time while attempting to name the individual, be that as it may, the recognition of the well-known face is typically prompt.

The unpredictability of a human face emerges from the consistent changes in the facial elements that occur after some time. In spite of these progressions, we people are still ready to perceive confronts and recognize the people. Obviously, our normal acknowledgment capacity stretches out past face acknowledgment, where we are similarly ready to rapidly perceive examples, sounds and smells. Lamentably, this common capacity does not exist in machines, consequently the requirement for falsely reenacting recognition in our endeavors to make canny independent machines.

Face recognition by machines can be priceless and has different vital applications, in actuality, for example, electronic and physical access control, national barrier and universal security. Mimicking our face recognition normal capacity in machines is a troublesome errand, yet not unthinkable. For the duration of our life time, numerous countenances are seen and put away normally in our recollections shaping a sort of database. Machine recognition of faces requires additionally a database which is generally constructed utilizing facial images, where some of the time distinctive face images of a one individual are incorporated to represent varieties in facial elements.

The usage of intelligent classifiers such as neural networks, support vector machine, and K-nearest neighbor etc,.. for the recognition of faces showed recently a higher efficiency and reliability than older techniques. This is due to the algorithm which these classifiers are based on; which is exactly a mimicking of how the humans recognize faces using their brains.

Current face recognition techniques depend on: identifying neighborhood facial features and utilizing them for face recognition or on universally breaking down a face in general. The primary methodology (neighborhood face recognition frameworks) utilizes facial components or features inside the face, for example, (eyes, nose and mouth) to relate the face with a man. The second approach (global face acknowledgment frameworks) utilizes the entire face for distinguishing the individual.

The improvement of intelligent frameworks that utilization neural systems is interesting and has of late pulled in more scientists into investigating the potential uses of such frameworks. Simulating the human discernments and demonstrating our faculties utilizing machines is extraordinary and may help mankind in therapeutic progression, space investigation, discovering elective vitality assets or giving national and global security and peace. Intelligent frameworks are by and large progressively created meaning to reenact our view of different inputs (examples, for example, images, sounds… and so forth. Biometrics is a case of famous applications for manufactured wise frameworks. The improvement of an intelligent face recognition framework requires giving adequate data and significant information amid machine learning of a face.

Recently, the Scale Invariant Feature Transform was proposed by (Lowe, 2004). The proposed algorithm was used as a feature descriptor and extractor of human faces. SIFT descriptor comprised a method for detecting interest points from a grey-level image at which statistics of local gradient directions of image intensities were accumulated to give a summarizing description of the local image structures in a local neighborhoods around each interest point, with the intention that this descriptor should be used for matching corresponding interest points between different images.

This algorithm was used by many researchers as a feature extractor in combination with intelligent classifiers such as neural network and SVM. The algorithm showed great efficiency in extracting the right features that distinguish human faces. Thus, the proposed system is a face recognition intelligent system based on SIFT algorithm for the feature extraction and backpropagation neural network for the classification. The purpose of this research is to evaluate the effectiveness of a backpropagation neural network in recognizing different faces and to compare the obtained results with those in the literature review. The developed framework consists of two main phases which are the processing phase and the classification phase in which the image is classified as different faces. In the image processing phase the face images are pre-processed using many techniques such as conversion to grayscale and filtering using median filter. Then the most significant technique takes place which is the feature extraction using SIFT. These techniques are done in order to enhance the quality of images and to extract the important features in such a way to take only the important face's features and ignoring the other features and parts of the image. At the end of this phase, the images are fed to a backpropagation neural network in which they are classified as different faces for different individuals.

1.1 Contributions of Research

- This thesis develops face recognition based SIFT and backpropagation neural network system, that has the capability of determining the human faces identities of presented faces of different individuals with different facial expressions.
- Moreover, within the work we propose a simple approach to extracting of 128 features from face using SIFT which reduces the processing and training time and also shows good recognition rate compared to other presented works.
- Within the work, we show the usefulness of using SIFT as a feature extractor of the face features using using artificial neural networks.

1.2 Aims of Thesis

The aim of the proposed system is to investigate the use of SIFT algorithm as a feature extractor of 128 features in combination with a backproagation neural network that learn these features and use them to generalize when some changes such as scale variance, different facial expression are induced. The aim of this research is to evaluate the effectiveness of the use of SIFT and backpropagation neural network together in recognizing different faces and to compare the obtained results with those in the literature review.

1.3 Thesis Overview

The rest of this thesis is divided into 8 chapters, which are structured as follows.

Chapter 1 is an introduction about the thesis. In this chapter, a definition of the thesis is presented; we set the aims, the contributions, and motivations. In addition, the structure overview of the thesis is discussed.

Chapter 2 introduces the literature review of face recognition systems in several aspects is presented. The problems that and drawbacks in the face recognition filed are also discussed. In addition, the various algorithms that were used for the face recognition are described in details.

Chapter 3 is a detailed and general explanation about the image processing. An introduction of the image processing is first presented. Then, we explain the image processing techniques and methods used in the medical field. We attempt to explain the used image processing methods of the proposed system in details.

Chapter 4 is a detailed explanation the artificial neural network where the concept and the various networks including the backpropagation neural network are explained.

Chapter 5 is a detailed explanation of the SIFT algorithm that is used in the proposed face recognition system.

Chapter 6 discusses the proposed system methodology, materials and methods are presented. The system flowchart and algorithm is presented in this chapter. Moreover, the methods used in order to come up with such system are discussed as well as the face images dataset used in training and testing the system.

Chapter 7 presents the classification stage of the developed system. It shows the learning and also testing phases of the system. The learning results are discussed in this chapter as well as the performance of the network in the testing stage.

Finally**, Chapter 8** shows the results comparison of the proposed face recognition based SIFT system are presented, discussed and compared with previously proposed systems of the same goal are explained in this chapter.

CHAPTER TWO
FACE RECOGNITION: A LITERATURE SURVEY

Face Recognition (FR) is an examination area traversing a few trains, for example, image preparing, pattern recognition, neural network classification system and computer vision. There are numerous utilizations of FR as appeared in Table 1. These recognition and classifications applications range from coordinating of photos to ongoing coordinating of reconnaissance video. Contingent upon the particular application, FRT has diverse level of trouble and requires extensive variety of methods. In 1995, an audit paper by (Chellappa et al., 2000) gives a through overview of FRT around then. Amid the previous couple of years, FRT is still under fast development.

2.1 The Challenges in FR

Despite the fact that numerous FR have been proposed, powerful face recognition is still troublesome. The late FERET test (Chaleppa et al., 2000) has uncovered that there are no less than two noteworthy difficulties:

- The illumination variety issue
- The posture variety issue

It is possible that one or both issues can bring about genuine execution debasement in a large portion of existing frameworks. Shockingly, these issues happen in numerous certifiable applications, for example, reconnaissance video. In the accompanying, I will examine some current answers for these issues.

The general face recognition issue can be defined as takes after: Given single picture or grouping of pictures, perceive the individual in picture utilizing a database. Taking care of the issue comprises of taking after strides: 1) face discovery, 2) face standardization, 3) ask database.

2.2 The Illumination Issue

Pictures of the same face show up contrastingly because of the adjustment in lighting. On the off chance that the change instigated by illumination is bigger than the distinction between people, frameworks would not have the capacity to perceive the information picture. To handle the illumination issue, scientists have suggested different strategies. It has been recommended that one can decrease variety by disposing of the most essential eigenface. What's more, it is confirmed in (Gorden, 1991) that disposing of the initial few eigenfaces appears to work sensibly well. Be that as it may, it causes the framework execution corruption for information pictures taken under frontal illumination.

In (Zhao et al., 2000) distinctive picture descriptions and separation measures are assessed. One vital conclusion that this research disadvantages is that none of these strategy is adequate without anyone else's input to conquer the illumination varieties. All the more as of late, another picture correlation strategy was proposed by (Jacobs et al., 2000). In any case this measure is not entirely illumination-invariant in light of the fact that the measure changes for a couple of pictures of the same item when the illumination changes.

An illumination subspace for a man has been built in (Phillipis et al., 2000) for an altered perspective point. In this manner under altered perspective point, recognition result could be illumination–invariant.

One downside to utilize this strategy is that we require numerous pictures per individual to develop the premise pictures of illumination subspace.

In (Ji, 2000) the creators recommend utilizing Principal Component Analysis (PCA) to tackle parametric shape-from-shading (SFS) issue. Their thought is entirely basic. They remake 3D face surface from single picture utilizing computer vision strategies. At that point process the frontal perspective picture under frontal illumination. Good results are illustrated. I will clarify their methodology in point of interest later. Really, there are a ton of issues in how to reproduce 3D surface from single picture.

We will examine two critical illumination-invariant FRT in the take after segments

2.3 The Pose Problem

The structure execution drops through and through when stance assortments are accessible in data pictures. On a very basic level, the present plan can be isolated into three sorts: 1) various pictures per individual are required in both get ready stage and affirmation stage, 2) different pictures per individual are used as a piece of get ready stage however emerge database picture per individual is open in affirmation stage, 3) single picture based systems. The second sort is the most surely understood one.

Distinctive pictures approaches: an edification based picture union methodology (Gordon, 1991) has been proposed for dealing with both stance and lighting up issues. This technique relies on upon lighting up cone to oversee light assortment. For assortments in light of turn, it needs to thoroughly resolve the GBR (summed up bas-help) instability while recreating 3D surface.

Hybrid methodologies: such countless of this write have been proposed. It is likely the most practical game plan up to now. Three reprehensive methodologies are examined in this report: 1) direct class based methodology (Zhao, 1999), 2) diagram planning based procedure (Zhao & Challeppa, 2000) 3) view-based eigenface technique (Beumier & Acheroy, 1999). This photo mix technique relies on upon the assumption of direct 3D object classes and expansion of linearity to pictures. In (sakamoto & Kriegnam, 1999) a healthy face affirmation arrangement in light of EBGM is proposed. They show liberal change in face affirmation under turn. Also, their strategy is totally modified, including face confinement, breakthrough recognition and graph planning arrangement. The disservice of this system is the need of accurate purpose of interest limitation which is troublesome when light assortments are accessible. The predominant eigenface approach has been acclimated to finish stance invariant. This strategy manufactures eigenfaces for each position. All the more starting late, a general system which is called bilinear model has been developed. The methodologies in this characterization have some essential burdens: 1) they require various pictures per individual to cover possible stances. 2) The light issue is disconnected from the stance issue.

2.4 Single Image Based Approaches

Gabor wavelet based on element extraction is suggested for the application of face recognition and is hearty to little point revolution. There are numerous papers on invariant components in computer vision writing. There are little written works looking at utilizing this innovation to face recognition. Late work in (Zhao, 1999) reveals some insight in this heading. For combining face pictures under various lighting or appearance. Because of its intricacy and calculation cost it is difficult to apply this innovation to face recognition.

2.5 The State of Art

In the accompanying areas, I will talk about some late research works in face recognition.

2.5.1 Applying shape-from-shading (SFS) to face recognition

The fundamental thought of SFS is to gather the 3D surface of item from the shading data in picture. With a specific end goal to gather such data, we have to expect a reflectance model under which the given picture is created from 3D object. There are numerous illumination models accessible. Among these models, the Lambertian model is the most well-known one and has been utilized broadly as a part of computer vision group for the SFS issue (Phillips et al., 2000) The nature of SFS makes it a not well postured issue as a rule. As it were, the reproduced 3D surface can't blend the pictures under various lighting edge. Luckily, hypothetical advances make SFS issue an all-around postured issue under specific conditions. The key equation in SFS problem is the following *irradiance equation*:

$$I[x,y] = R(p[x,y], q[x,y]) \tag{2.1}$$

where $I[x,y]$ is the image, R is the reflectance outline are and $p[x,y], q[x,y]$ are the shape angles (fractional subordinates of the profundity map).With the presumption of a Lambertian surface and a solitary, far off light source, the condition can be composed as takes after:

$$I = \rho \cos\theta$$

or

$$I = \rho \frac{1 + pP_s + qQ_s}{\sqrt{1+p^2+q^2}\sqrt{1+P_s^2+Q_s^2}} \tag{2.2}$$

Since SFS count gives face shape information, illumination and position issues can be settled at the same time. For example, we can deal with the illumination issue by rendering the model picture Ip from a given data picture I. This ought to be conceivable in two phases: 1) apply SFS count to get (p,q), 2) the new deliver the model picture Ip under lighting point $\alpha = 0$.

To survey some current SFS computations, (Zhao, 1999) applies a couple SFS estimations to 1) designed face pictures which are made in light of Lambertian model and enduring albedo, 2) honest to goodness face pictures. The test comes to fruition exhibit that these estimations are adequately awful for certified face pictures with the ultimate objective that an enormous change in face recognition can be refined. The reason is that face is made out of materials with different reflecting properties: cheek skin, lip skin, eye, thus on subsequently, Lambertian model and steady albedo can not give awesome assessment. The authors in (Zhao et al., 2000) develop a symmetric SFS figuring using the Lambertian and moving albedo $\rho(x,y)$ as an unrivaled alternative. With the aid of a non-particular 3D head model, they can condense the two-phase system of procuring model picture (1. input picture to shape by method for SFS, 2. shape to model picture) to one phase: input picture to model picture clearly.

Their estimation is associated with more than 150 face pictures from the Yale University and Weizmann database. The results clearly demonstrate the unrivaled way of model pictures rendered by their technique. They in like manner lead three tests to survey the effect in recognition execution when their computation is joined with existing FRT. The key test shows the adjustments in recognition execution by using the new illumination-invariant measure they portray. The results are showed up in Table 2 and Table 3. The second examination shows that using the rendered model pictures as opposed to novel data pictures can basically improve existing FRT, for instance, PCA and LDA.

Table 1: Different measures performance for 3 images per person

Database	Image Measure	Gradient Measure	Illumination-Invariant Measure
Yale	68.3%	78.3%	83.3%
Weizmann	86.5%	97.9%	81.3%

Table 2: Three different measures performance for 2 images per person

Database	Image Measure	Gradient Measure	Illumination-Invariant Measure
Yale	78.3%	88.3%	90.0%
Weizmann	72.9%	96.9%	87.9%

Table 3: Performance with/without using prototype image

Database	PCA	LDA	P-PCA	P-LDA
Yale	71.7%	88.3%	90.0	95.0%
Weizmann	97.9%	100%	95.8%	98.9%

2.5.2 Applying illumination cone to face recognition

In prior work, it is demonstrated that the pictures under subjective mix of light sources shape a got cone up picture space. This cone, called light cone, can be made from as few as three pictures. Figure 1 exhibits the course toward building the light cone. Figure 1a show seven noteworthy pictures with various light utilized as a bit of estimation of brightening cone. Figure 1b shows the reason pictures of light cone. They can be utilized to make pictures under discretionary enlightenment condition. Figure 1c shows the joined pictures from brightening cone of one face.

The repeated 3D face surface and enlightenment cones can be joined to combine pictures under various brightening and position. In (Georghiades et al., 2001) the authors use earlier information about the state of face to choose the Generalized bas-help (GBR) (Beumier & Kriegnam, 1998) indefinite quality. Once the GBR parameters are figured, it is a crucial matter to render outlined pictures under various enlightenment and position. Figure 2.1 shows the redid face surface. Figure 2.2 shows the composed pictures of a face under various position and enlightenment. Note that these pictures are made from the seven arranging pictures in Figure 1.a where the position is settled and just little grouping in enlightenment. Inquisitively, the manufactured

pictures show clearing arrangement in position furthermore in light. They performed two blueprints of acknowledgment examinations. The focal test, where just brightening shifts while position stays settled, was wanted to adjust other acknowledgment calculations with enlightenment cone strategy. There are a total of 450 pictures (45 brightening conditions × 10 faces). These pictures are separated into for social events (12°, 25°, 50° and 77°) as indicated by the edge between light source and camera focus point. Table 5 displays the outcomes.
Cones-connected implies that illumination cone was developed without cast shadow and

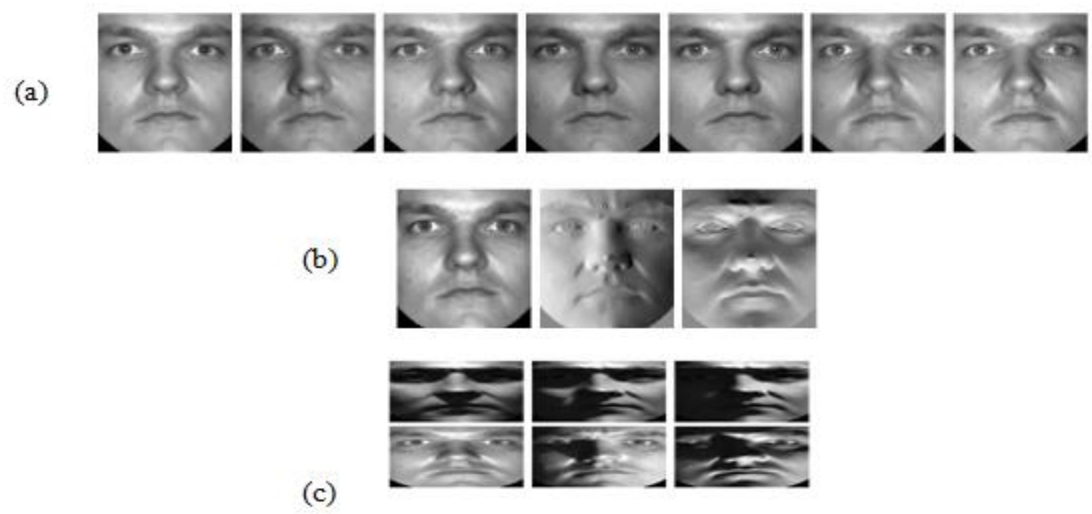


Figure 2.1: The process of constructing illumination cone (Georghiades et al., 2001)

Figure 2.2: Face reconstruction of 10 persons (Georghiades et al., 2001)

cones-cast implies that the reproduced face surface was utilized to decide cast shadow. Notice that the cone subspace estimation has the same execution as the first enlightenment cone.

Figure 2.3: Face synthesizing images (Georghiades et al., 2001)

Table 4: The error rate of different illumination with a fixed pose (Georghiades et al., 2001)

EXTRAPOLATION IN ILLUMINATION			
	Error Rate(%) vs. Illum		
Method	**Subset 2**	**Subset 3**	**Subset 4**
Correlation	0.0	23.3	73.6
Eigen faces	0.0	25.8	75.7
Eigen faces w/o 1 st 3	0.0	19.2	66.4
Linear- Subspace	0.0	0.0	15.0
Cones – attached	0.0	0.0	8.6
Cones – east (Subspace Approx.)	0.0	0.0	0.0
Cones – east	0.0	0.0	0.0

In the second test, they are surveying the recognition execution under assortment in stance and illumination. There is a whole of 4,050 images (9 stances × 45 illumination conditions × 10 faces). Figure 2.4 exhibits the results. Their figuring has low botch rate for all stances except for on the convincing lighting condition.

We can make the going with conclusions from their test comes to fruition: 1) we can achieve stance/illumination invariant recognition by using minimal number of images with changed position and to some degree differing illumination, 2) the images of the face exposed to different and variable illumination might be all around approximated by a low-dimensional subspace.

2.5.3 Linear object classes method

Consider the issue of seeing a face with different positions and appearances when developing an image is given. Human visual framework is determinedly arranged to play out this try. The conceivable reason is that we manhandle the earlier data about how go up against pictures change. Thusly, the thought here is to take in picture change from cases and after that apply it to the new face picture keeping in mind the end goal to join the virtual perspective that can be utilized as a bit of existing face acknowledgment structure. Poggio and Vetter (Poggio &Vetter, 2002) present the course of action of making fake new pictures of a thing. Their work depends on upon the probability of straight question classes. These are 3D contradicts whose 3D shape can be tended to as quick mix of to some degree number of model things. Thusly, if the representation set contains frontal and pivoted viewpoint pictures, we can blend pictures of turned perspective from the given information picture.

For human-made articles, which as often as possible contain cuboids, barrels, or other geometric primitives, the suspicion of straight question classes appears, in every way, to be trademark. Regardless, by temperance of face, it is not clear what number of cases is adequate. They test their reasoning on a game-plan of 50 faces, every given in two presentations (22.5˚ and 0˚). In their test, one face is picked as test face, and the other 49 countenances are utilized as outlines. In Figure 5, every test face is appeared on the upper left and the joined picture is appeared on lower right. The ensured turned test face is appeared on the lower left. In the upper right, they likewise show the blend of the test face through 49 cases in test presentation.

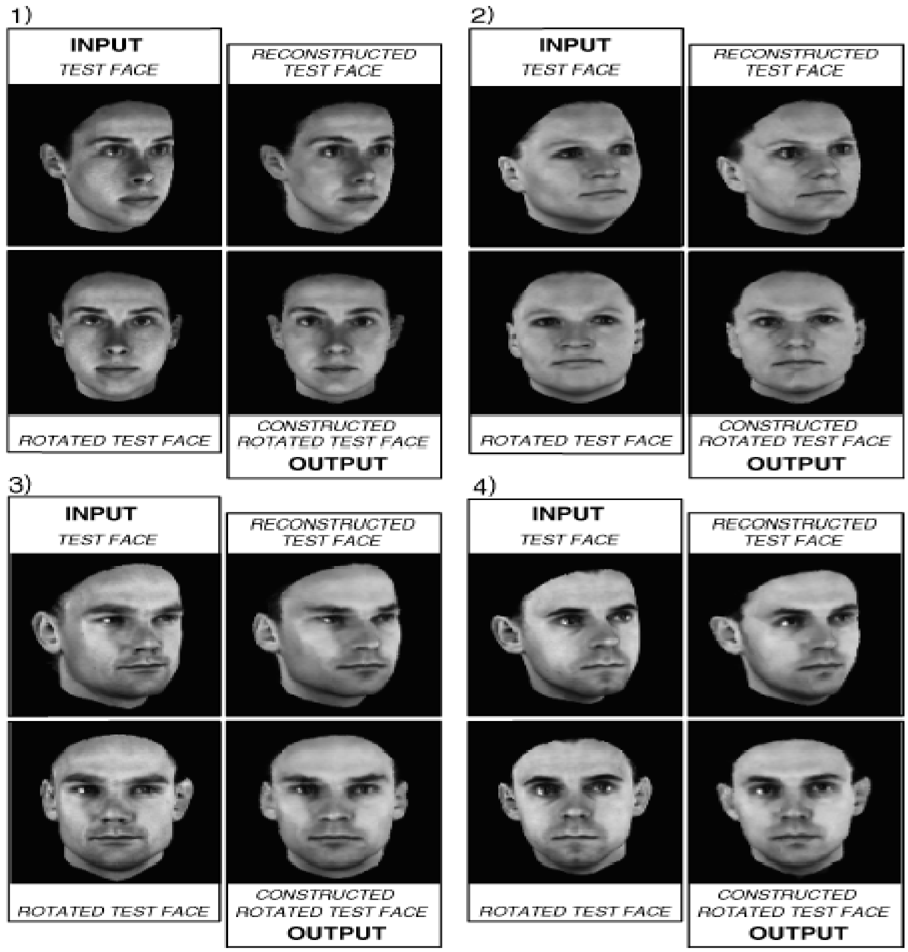


Figure 2.4: The face is rotated by using 49 faces as examples (not illustrated) and the result are marked as output (Poggio & Vetter, 2002)

This diversion of the test face should be appreciated as the projection of the test face into the subspace crossed the other 49 outlines. The results are not immaculate, yet rather considering the little size of delineation set, the entertainment is entirely awesome. All things considered, the likeness of the multiplication to the data test face licenses us to figure that an outline set of hundreds faces may be satisfactory to build up a massive variety of different appearances. We can assume that the immediate inquiry class approaches maybe a classy assessment, despite for complex things as appearances.

In this manner, given just a solitary face image, we can create extra engineered face images under various perspective point. For face recognition assignment, these manufactured images could be utilized to handle the stance variety. Furthermore, this methodology does not require any profundity data, so the troublesome strides of producing 3D models can be stayed away from.

2.5.4 View-based eigenspace

The eigenface procedure for Turk and Pentland (Turk &Pentland, 1991) was summed up to see based eigenspace system for managing position grouping. These improvements addressed arrangement in position and provoke a more strong acknowledgment structure.

They detail the issue of face acknowledgment under various positions as takes after: given N people under M arranged positions, one can make a "perspective based" game-plan of M separate eigenspaces. Each eigenspace gets the arrangement of N people in a typical position.

In connection based method, the hidden step is to pick the position of information face picture by selecting the eigenspace which best portrays it. This could be expert by figuring the Euclidian parcel between information picture and the projection of information picture in each eigenspace. The eigenspace yielding the littlest separation is the one with most for all intents and purposes indistinguishable position to information picture. Once the best eigenspace is resolved, the information picture is coded utilizing the eigenfaces of that space and after that clear.

They have assessed the perspective based system with 189 pictures, 21 individuals with 9 positions. The 9 positions of every individual were reliably disconnected from - 90° to 90° along the even plane. Two specific test mythologies were utilized to judge the acknowledgment execution.

In the basic strategy of examinations, the incorporation execution was endeavored by method for get prepared on the subset of accessible perspective {±90°, ±45°, 0°} and testing on the broadly engaging sees {±68°, ±23°}. The conventional acknowledgment rate was 90% for perspective based methodology. A second game-plan of trials test the extrapolation execution by method for anticipating a degree of accessible perspective {e.g., - 90° to +45°} and testing on perspectives outside the course of action reach {e.g., +68°, +90°}. For testing positions isolated by ±23° from the arranging go, the common acknowledgment rate was 83% for perspective based technique.

2.5.5 Curvature-based face recognition

In (Haider & Kaneko, 2000) they use back and forth movement of surface to perform face recognition. This is an inconceivable thought since the estimation of curve at a point at first look is invariant under the assortment of viewpoint and illumination. In this approach, a turn laser scanner produces data of adequately high assurance with the true objective that exact curve calculation can be made. Face division can be made in perspective of the sign of Gaussian curve; this licenses two surface sorts: bended/depressed and saddle zones. Their surface segment extraction contains twist sign and first back and forth movement, key bearing, umbilic centers and extremes in both essential recurring patterns. The most amazing and minimum twist at a point describes the essential shapes. The headings associated with fundamental shapes are the focal direction. The vital rhythmic movements and the crucial heading are given by the eigenvalues and eigenvectors of shape framework. The aftereffect of two principal curves is Gaussian recurring pattern. Besides, mean back and forth movement is described by the mean estimation of two vital recurring patterns.

For all intents and purposes, in light of the way that these rhythmic movement estimations contain second demand midway auxiliaries, they are incredibly fragile to hullabaloo. A smoothing channel is required before figuring curve. Here, the bind is the best approach to pick an appropriate smoothing level. If the smoothing level is too low, twice subordinate will build noise with the ultimate objective that rhythmic movement estimation is worthless. On the other hand, over smoothing will modify the surface components we are endeavoring to measure. In their use, they precompute the shape values using a couple of extraordinary levels of smoothing. They use the back and forth movement maps from low smoothing level to develop the zone of components. By then, use the prior data of face structure to pick the curve values from the precomputed set. This is done physically, I think. An instance of focal recurring pattern maps is given in figure 2.6.Segmentation is to some degree clear. By using the sign of Gaussian curve, K, and mean shape, H, face surface can be apportioned into four different sorts of regions: K+, H+ is bended, K+, H-is internal, K-, H+ is seat with and K-, H-is seat with. The point of confinement of these areas is called illustrative twist where Gaussian curve is zero. Figure 2.6 shows a case.

The author similarly talks about the calculation of surface descriptors. She tries to find however much information as could be normal from the achieved data with the true objective that this information is as stand-out as the individual.

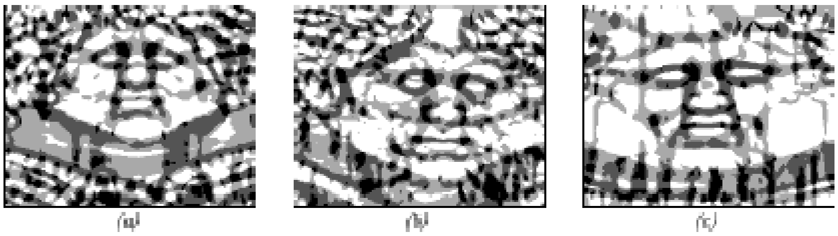


Figure 2.5: This shows 3 segmented faces using the sign Gaussian and mean curvature (Haider & Kaneo, 2000)

With such a rich arrangement of data accessible, there are numerous approaches to build an examination technique. The creator utilizes highlight extraction and layout coordinating to perform face recognition. In the analysis, test set comprises of 8 countenances with 3 sees each. For every face there are two forms without demeanor and one with appearance. The investigation comes about demonstrate that 97% of the examinations are right.

As I would like to think, the upsides of ebb and flow based procedure are: 1) it takes care of the issue of stance and illumination variety ate the same time. 2) There is a lot of data in ebb and flow map which we haven't exploited. It is conceivable to locate a proficient approach to manage it.

Be that as it may, there are some inborn issues in this methodology: Laser range discoverer framework is considerably more costly contrasted and camera. What's more, this method can't be connected to the current image database. This makes individuals would prefer not to pick it in the event that they have another decision. Even however the fury discoverer is not an issue any more, the calculation expense is too high and the ebb and flow count is exceptionally touchy to commotion. On the off chance that we utilize vital part investigation to manage range information, the mistake rate likely will be comparative while the calculation many-sided quality is much lower. We can develop 3D face surface from 2D image rather than costly range discoverer. There are a ton of calculations accessible. However, you won't have the capacity to figure arch from reproduced 3D face surface. As specified before, shape count includes second subordinate of surface. Just the high-determination information, for example, laser range discoverer makes the precise arch figuring conceivable.

2.5.6 3D model-based face recognition

To lessen the expense of framework, Beumier and Acheroy (Beumier & Acheroy, 1999) pick the 3D securing framework comprising of standard CCD camera and organized light. It depends on the projection of a known light pattern. The pattern disfigurement contains thee profundity data of the article. 3D surface remaking is finished by stripe recognition and marking. Shape every purpose of a stripe and its name, triangulation takes into account X, Y, Z estimation. This procedure is quick while offering adequate determination for recognition reason.

There are 120 people in their test. Each on is taken three shots, relating to focal, constrained left/right pivot and up/down turn. Programmed database utilizes the programmed system to get 3D data of every person. In manual database, the 3D extraction procedure was performed by clicking starting focuses in the distorted pattern.

With the 3D remaking, they are searching for attributes to diminish the 3D information to an arrangement of elements that could be effectively and immediately looked at. In any case, they observed nose is by all accounts the main strong component with constrained exertion. Along these lines, they surrendered highlight extraction and considered worldwide coordinating of face surface.

15 profiles are extricated by the crossing point of face surface and parallel plane separated with 1 cm. A separation estimation called profile separation is characterized to measure the distinction between 3D surfaces. This methodology is moderate: around 1 second to look at two face surfaces. Keeping in mind the end goal to accelerate this calculation, they attempted to utilize just the focal profile and

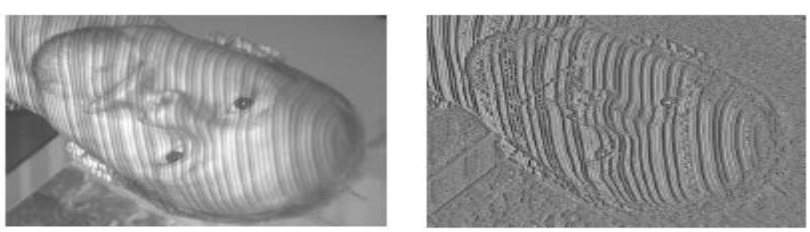


Figure 2.6: The pattern deformation

Figure 2.7: Reconstructed face surface (Beumier & Acheroy, 1999)

two lateral profiles in comparison. ROC bends are appeared in figure 10 to delineate the impact of correlation system. In focal/sidelong profile examination, blunder rate is relinquished (from 3.5% to 6.2%) to gain the velocity of surface correlation. In the left of figure 10, the manual refinement gives us better recognition execution. This lets us know that there is space to enhance programmed 3D obtaining framework.

Advantage: 1) extra cost is just the projector and pattern slide. 2) Switching the slide on and off permits gaining both 2D image and 3D data. The combination of 2D and 3D data can expand the recognition execution. 3) The projector illumination diminishes the impact of surrounding light. 4) 3D reproduction and profile correlation can maintain a strategic distance from stance variety.

Issues: 1) programmed 3D reproduction is sufficiently bad. A conspicuous change should be possible by manual refinement. 2) Profile coordinating is extremely costly computational assignment. In face confirmation, this is not an issue. However, in face recognition with enormous database, the rate would be horrendously moderate.

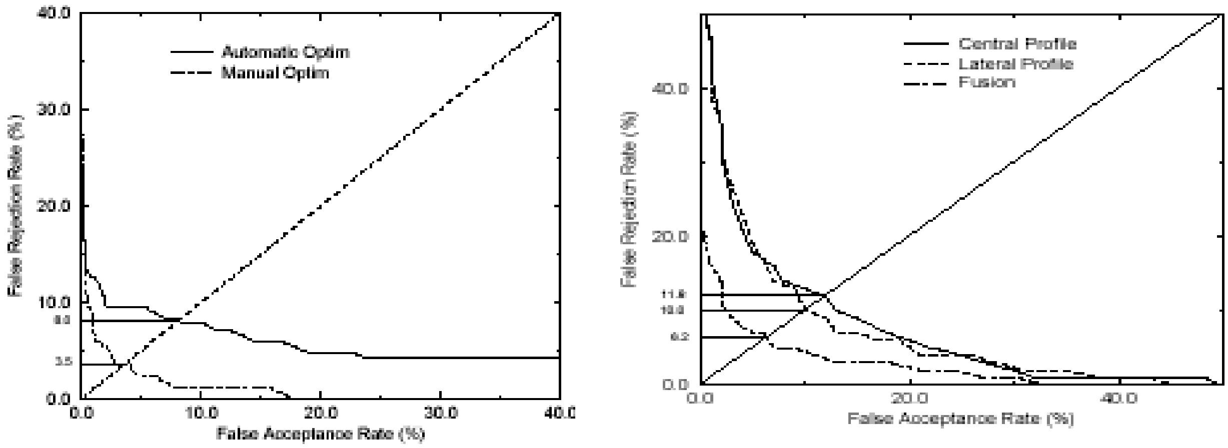


Figure 2.8: The ROC curves of 3D face surface recognition

2.5.7 Elastic bunch graph matching

In (Belhumeur & Kriegnam, 1998) they utilize Gabor wavelet change to concentrate face highlights so that the recognition execution can be invariant to the variety in postures. Here, I need to discuss a few phrasings they utilize first and talk about how they fabricate the face recognition framework.

For every element point on the face, it is changed with a group of Gabor wavelets. The arrangement of Gabor wavelets comprises of 5 diverse spatial frequencies and 8 introductions. In this manner, one component point has 40 comparing Gabor wavelet coefficients. A plane is characterized as the arrangement of Gabor wavelet coefficients for one component focuses. It can be composed as.

A marked Graph G speaks to a face comprises of N hubs associated by E edges. The hubs are situated at highlight focuses called fiducial focuses. For instance, the students, the sides of mouth, the tip of nose are all fiducial focuses. The hubs are marked with planes. Charts for various head posture contrast in geometry and neighborhood highlights. To have the capacity to analyze charts of various represents, the physically characterizes pointers to relate comparing hubs in various diagrams.

With a specific end goal to concentrate diagrams consequently for new face, they require a general representation for face. This representation ought to cover an extensive variety of conceivable varieties in appearance of face. This delegate set has stack-like structure, called face

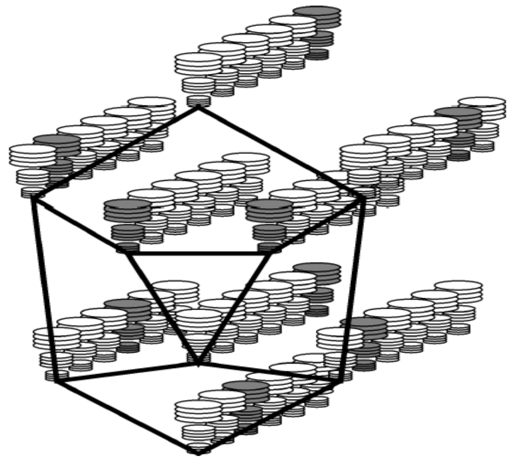


Figure 2.9: The Face Bunch (Belhumeur & Kriegnam, 1998)

cluster diagram (FBG) (see Figure 2.9).

An arrangement of planes alluding to on fiducial point is known as a cluster. An eye group, for case, may incorporates planes from shut, open, female and male eyes and so on to cover conceivable variety. The Face Bunch Graph is given the same structure as the individual diagram.

In hunting down fiducial focuses in new image of face, the method portrayed underneath chooses the best fitting plane from the cluster devoted to each fiducial point.

The main arrangement of charts is created physically. At first, when the FBG contains just few confronts, it is important to check the coordinating result. Once the FBG is sufficiently rich (roughly 70 diagrams), the coordinating results are really great.

Coordinating a FBG on another image is finished by expanding the diagram closeness between image chart and the FBG of the same posture. For an image diagram G with hubs n = 1,… ,N and edges e = 1,… ,E and FBG B with model chart m = 1,… ,M the comparability is characterized as

$$S(G,B) = \frac{1}{N}\sum_{n}\max_{m}(S_{\phi}(J_{n}^{I}, J_{n}^{B_m})) - \frac{\lambda}{E}\sum_{e}\frac{(\vec{x}_e^{I} - \vec{x}_e^{B})^2}{(\vec{x}_e^{B})^2}$$

Since the FBG gives a couple planes to each fiducial point, the best one is picked and used for examination. The best fitting planes serve as neighborhood authorities for the new image.

They use the FERET database to test their system. Regardless, the size and territory of face is determined and go up against image is institutionalized in size. In this movement a couple FBGs of different size are required; the best fitting one is used for size estimation. In FERET database, each image has an imprint showing the position, there is no convincing motivation to gage stance. Notwithstanding, stance could be assessed actually in tantamount course as size.

In the wake of expelling model outlines from the presentation images, recognition is possible by standing out an image graph from each and every model chart and selecting the one with most vital resemblance regard. An examination against a showcase of 250 individuals takes shy of what one second.

The positions used here are: objective frontal point of view (fa), frontal viewpoint with different expression (fb), half-profile right (hr) or left (hl), and profile right (pr) and left (pl). Recognition results are showed up in Table 6.

The recognition rate is high for frontal against frontal images (first segment). This is a direct result of the way that two frontal points of view demonstrate simply little assortment. The recognition rate is till high for right profile against left profile (third line). Exactly when taking a gander at left and right half-profile, the recognition rate drops altogether (second segment). The possible reason is the assortment thusly point – visual examination exhibits that insurgency edge may change by up to 30°. By differentiating frontal viewpoints or profile against half profile, a further reduction in recognition rate is viewed.

From the test occurs, obviously Gabor wavelet coefficients are not invariant under turn. Before performing recognition, notwithstanding all that you need to gage stance and discover looking at FBG.

Table 5: Recognition comes about for cross-keep running between various exhibitions.

Recognition Results For Cross-Runs Between Different Galleries							
Model Gallery		Probe Images		First Rank		First 10 Ranks	
				#	%	#	%
250 fa		250 fb		245	98	248	99
250 hr		181 hl		103	57	147	81
250 pr		250 pl		210	84	236	94
249 fa	1 fb	171 hl	79 hr	44	18	111	44
171 hl	79 hr	249 fa	1 fb	42	17	95	38
170 hl	80 hr	217 pl	33 pr	22	9	67	27
217 pl	33 pr	170 hl	80 hr	31	12	80	32

2.5.8 Experiment

Their examination was done utilizing face pictures of 50 people. Every individual gives six facial pictures view point and expression grouping. Some of these pictures are picked as the preparation pictures and the rest are taken as test picture. Thusly, the course of action set and testing set are disjoint.

For another test picture, subsequent to compelling the portion focuses, a 36×9×4-estimation shape vector and a 40×10-estimation surface vector are resolved. These two vectors are normal into taking a gander at subspace. The projection coefficients are stretched out to diagram a composed section vector.

Keeping in mind the end goal to assess acknowledgment execution, two examinations are performed with various segments, different number of supervisor parts and specific classifiers.

Case 1: Comparison of acknowledgment execution with various fragments, point signature (PS), Gabor coefficients (GC) and PS+GC.

Figure 14 (a) shows the acknowledgment rate versus subspace estimations with various picked highlights. The outcomes affirm their supposition that mix of 2D and 3D data can redesign acknowledgment execution.

Case 2: look at the acknowledgment execution of various classifiers, closeness utmost and Support Vector Machine. Figure 2.12 (b), (c) and (d) demonstrate the acknowledgment rate versus subspace estimation with various classifiers. Result in (b) is picked up utilizing point signature as highlight, (c) is gotten utilizing Gabor coefficients as highlight and (c) is obtained utilizing PS+GC as highlight. With SVM as the classifier, higher acknowledgment rate is acquired in every one of the three cases.

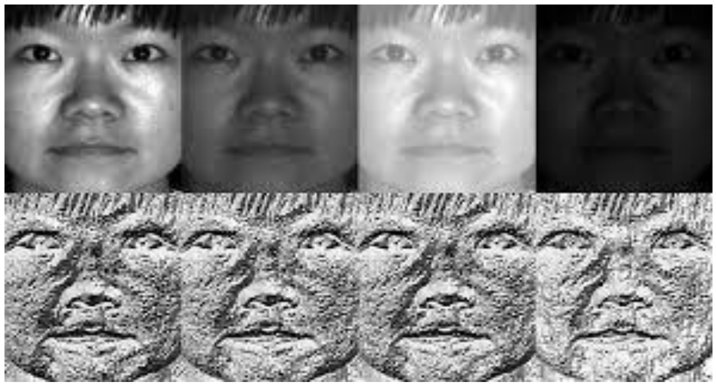


Figure 2.10: Recognition rate vs. subspace dimensions (Ji, 2000)

- **Pose estimation from single image**

By and large, a face recognition issue can be isolated into two noteworthy parts: standardization and recognition. In standardization, we have to evaluate the size, illumination, demeanor, and posture of face from the given image and afterward change input image into standardized organization which can be perceived by the recognition calculation. Subsequently, how to gauge posture precisely and productively is a vital issue in face recognition. Tackling this issue is a key stride in building a strong face recognition framework.

(Ji, 2000) propose another methodology for assessing 3D posture of face from single image. He accept that the state of face can be approximated by a circle. The stance of face can be communicated as far as yaw edge, pitch point and move edge of oval (see Figure 2.13, 2.14). His framework comprises of three noteworthy parts: understudy identification, face recognition and posture estimation.

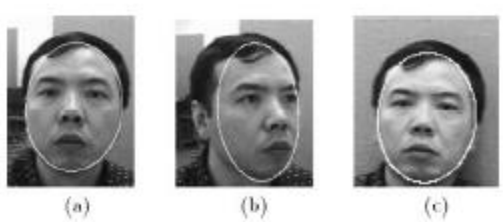


Figure 2.11: Ellipse approximation of a face (Ji, 2000)

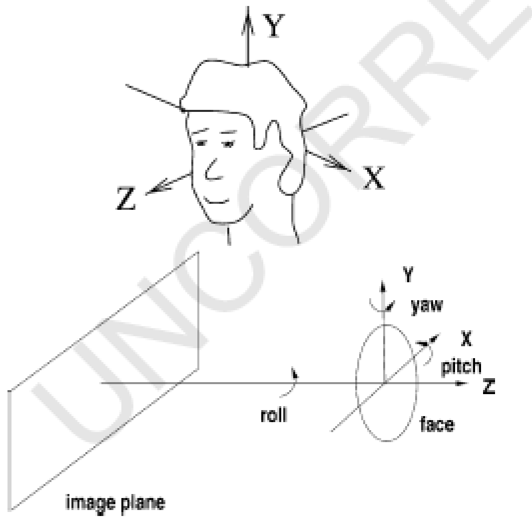


Figure 2.12: Yaw, pitch and roll angle poses of a face (Ji, 2000)

CHAPTER 3
IMAGE PROCESSING PRINCIPLES

Image processing is a very important aspect of pattern recognition and machine learning field. It offers various techniques to manipulate image data, feature extraction, image enhancement, and image segmentation. Image manipulation techniques include image sampling for up-scaling or down-scaling, conversion to gray images, black and white, etc.

Imaging frameworks develop a (yield) image in light of (info) signs from various sorts of articles. They can be grouped in various ways, e.g. as indicated by the radiation or field utilized, the property being researched, or whether the images are formed straightforwardly or by implication. Medicinal imaging frameworks, for instance, take input signals which emerge from different properties of the body of a patient, for example, its lessening of x-beams o impression of ultrasound. The subsequent images can be ceaseless, i.e. simple, or discrete i.e. advanced; the previous can be changed over into the last by digitization. The test is to get a yield image that is a precise representation of the information sign, and afterward to break down it and concentrate however much analytic data from the image as could be expected (Warfield, et al., 1998).

3.1 Principles of Image Processing

A complete advanced image processing framework (Figure. 3.1) is a gathering of equipment (gear) and programming (computer programs) that can:

(i) gain an image, utilizing proper sensors to distinguish the radiation or field and catch the elements of enthusiasm from the item in the most ideal way. On the off chance that the identified image is constant, i.e. simple, it should be digitized by a simple to-computerized converter (ADC); (ii) store the image, either incidentally in a working image store utilizing read/compose memory gadgets known as arbitrary access memory (RAM) or, all the more for all time, utilizing attractive media (e.g. floppy circles or the computer hard plate memory), optical media (e.g. Cd ROMs or DVDs) or semiconductor innovation (e.g. streak memory gadgets); (iii) control, i.e. process, the image; and (iv) show the image, in a perfect world on a TV or computer screen, which contains lines of persistently shifting, i.e. simple, force. This requires the generation of a simple video show signal by an advanced to-simple converter (DAC).

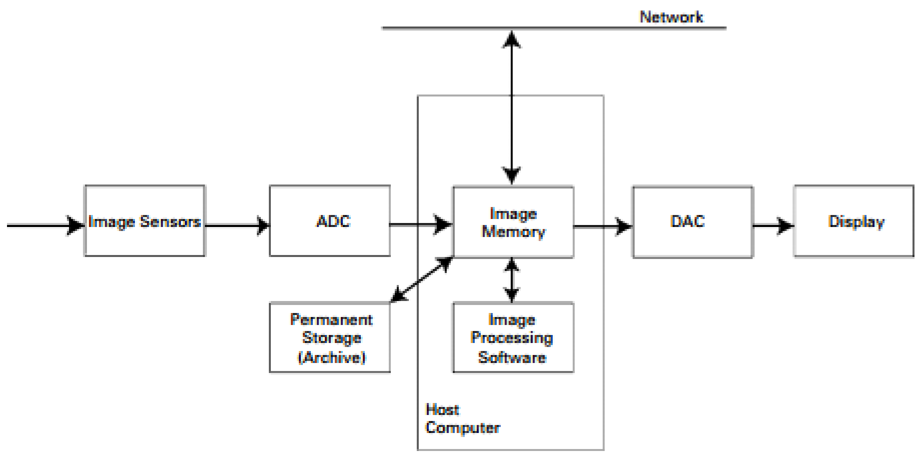


Figure 3.1: Digital image processing system

3.2 Image Enhancements

Image upgrade brings about an image which either looks better to an onlooker, a subjective marvel, or which performs better in an ensuing processing class. Upgrade may include conforming the splendor of the image, on the off chance that it was excessively dim or too splendid, or its difference, if for instance it contained just a couple shades of dark, giving it a washed-out appearance. On the other hand, it may include smoothing an image that contains a ton of commotion or dot, or honing an image so that edges inside it are all the more effectively seen.

Images are frequently altogether debased in the imaging framework, and image rebuilding is utilized to switch this corruption. This would incorporate switching the impacts of: uneven illumination, non-straight identifiers which create a yield (reaction) that is not corresponding to the info (boost), contortion, e.g. "pincushion" and "barrel" bends brought on by ineffectively centering focal points or electron optics, development of the article amid obtaining, and undesirable commotion (Figure 3.2). The way to image rebuilding is to show the debasement and afterward to utilize an opposite operation to turn around it (Fan et al., 2002).

There exist numerous methods that can upgrade an advanced image without ruining it. The improvement techniques can extensively be isolated into the accompanying two classifications:

1. Spatial Domain Methods.

2. Recurrence Domain Methods.

In spatial area systems, we specifically manage the image pixels. The pixel qualities are controlled to accomplish fancied improvement. In recurrence area techniques, the image is initially moved into recurrence space. It implies that, the Fourier Transform of the image is registered first. All the upgrade operations are performed on the Fourier change of the image and afterward the Inverse Fourier change is performed to get the resultant image. These upgrade operations are performed keeping in mind the end goal to alter the image brilliance, contrast or the dissemination of the dim levels. As a result the pixel esteem (forces) of the yield image will be altered by change capacity connected on the information values (Gonzalez & woods, 2001).

Image upgrade just means, changing an image f into image g utilizing T. (Where T is the change. The estimations of pixels in images f and g are signified by r and s, individually. As said, the pixel values r and s are connected by the expression,

$$s=T(r) \tag{3.1}$$

Where T is a change that maps a pixel esteem (r) into a pixel esteem. The aftereffects of this change are mapped into the dim scale range as we are managing here just with dim scale computerized images.

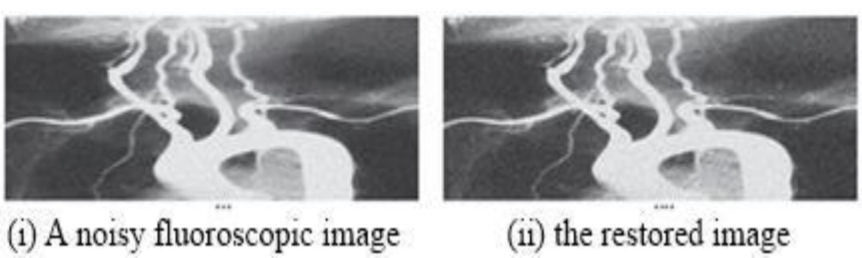


Figure 3.2: Image restoring

3.2.1 Contrast adjustments

Regularly, images have a low element reach and a large number of its elements are hard to see. We will exhibit diverse force changes that will enhance the presence of the images. Enhancing the presence of an image does not just serve a stylish part – frequently, it can enhance the execution of image division calculations and highlight recognition.

Amid difference conformity, the force estimation of every pixel in the crude image is changed utilizing an exchange capacity to shape a complexity balanced image. The most widely recognized exchange capacity is the gamma contrast conformity:

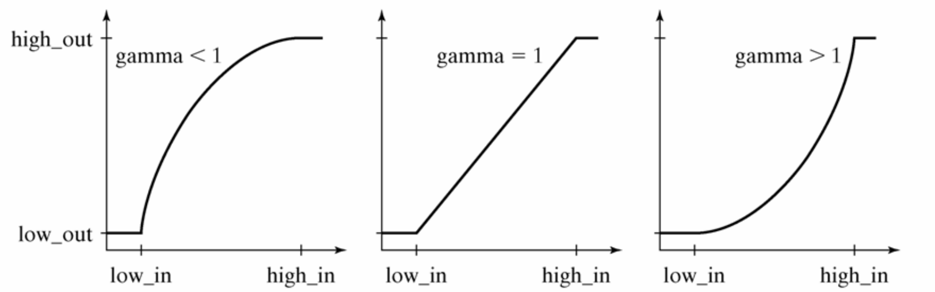


Figure 3.3: Gamma correction (Gonzalez & woods, 2001)

Here **low_in** and **low_high** give the low and high grayscale intensity values for the contrast adjustment, and **gamma** gives the exponent for the transfer function.

3.3 Data Compression and Data Redundancy

Image compression decreases the measure of information expected to depict the image. Images require huge record sizes, e.g. those involving 512×512 pixels require around 1/4 MB of space, practically identical to an archive containing 40 pages of content. The compression lessens the document measure so that the image can be all the more effectively put away or transported electronically, by means of communication for instance, in a shorter time. Pressure is conceivable on the grounds that images have a tendency to contain excess or redundant data. Elective stockpiling plans can store the data all the more successfully, i.e. in littler records, and decompression calculations can be utilized to recover the first image information. On the off chance that every one of the information is safeguarded in the packed document, though with various coding, the compression is lossless; this is compulsory for restorative images. Littler image records (i.e. more noteworthy compression can be acquired with lossy compression procedures, which don't protect the majority of the information of the first image, yet by and by keep up an image of adequate quality.

There are distinctive techniques to manage various types of previously mentioned redundancies. Accordingly, an image compressor regularly utilizes a multi-step calculation to decrease these redundancies.

3.3.1 Compression methods

Amid the previous two decades, different compression strategies have been created to address significant difficulties confronted by computerized imaging (Wallace, 1991). These compression techniques can be ordered comprehensively into lossy or lossless compression. Lossy compression can accomplish a high compression proportion, 50:1 or higher, since it permits some satisfactory corruption. However it can't totally recoup the first information. Then again, lossless compression can totally recuperate the first information however this lessens the compression proportion to around 2:1. In medicinal applications, lossless compression has been a prerequisite since it encourages exact conclusion because of no corruption on the first image. Moreover, there exist a few lawful and administrative issues that support lossless compression in medicinal applications.

- Lossy Compression Methods

By and large most lossy compressors (Figure 3.4) are three-stage calculations, each of which is as per three sorts of excess said above.

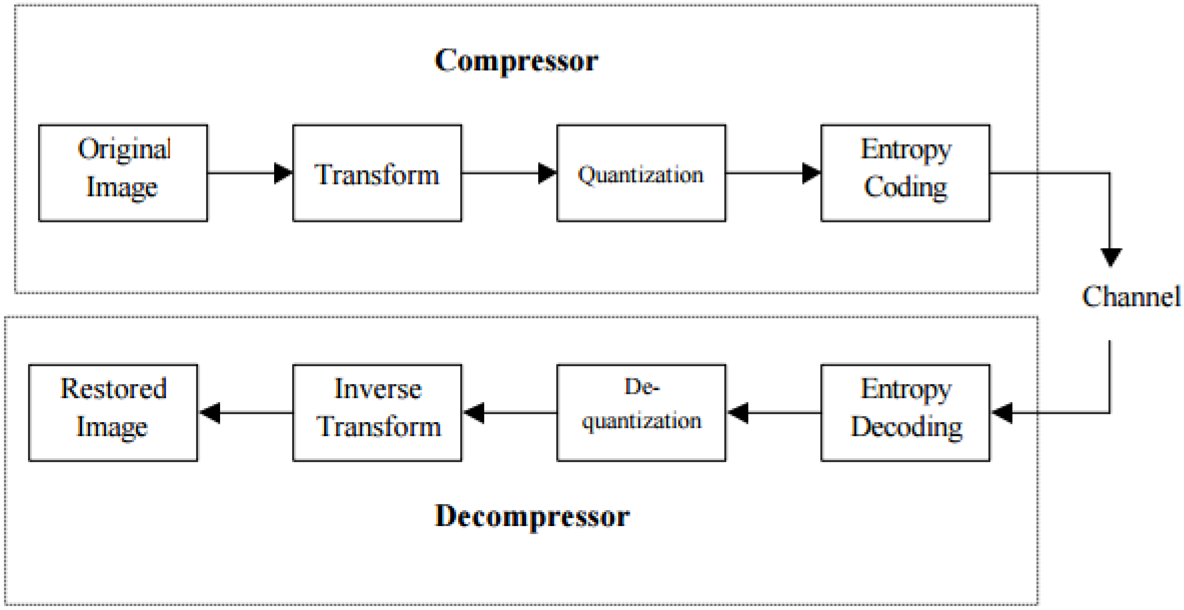


Figure 3.4: Lossy compression (Wallace, 1991)

The primary stage is a change to dispense with the between pixel repetition to pack data proficiently. At that point a quantizer is connected to expel psycho-visual excess to speak to the stuffed data with as The first stage is a transform to eliminate the inter-pixel redundancy to pack information efficiently. Then a quantizer is applied to remove psycho-visual redundancy to represent the packed information with as few bits as possible. The quantized bits are then efficiently encoded to get more compression from the coding redundancy.

- Lossless Compression Methods:

Lossless compressors (Figure 3.5) are usually two-step algorithms. The first step transforms the original image to some other format in which the inter-pixel redundancy is reduced. The second step uses an entropy encoder to remove the coding redundancy. The lossless decompressor is a perfect inverse process of the lossless compression of bits as could be expected under the circumstances. The quantized bits are then effectively encoded to get more compression from the coding repetition.

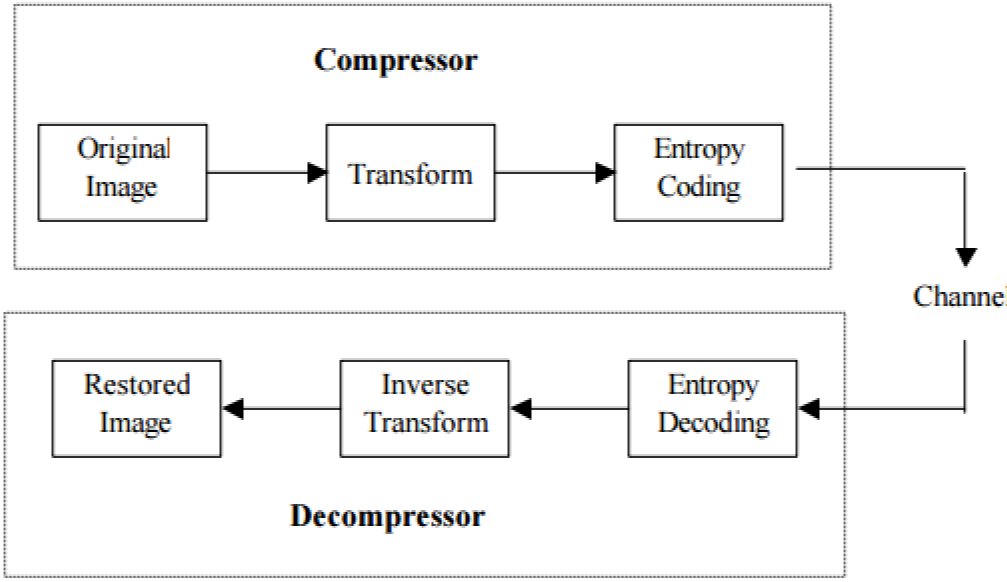


Figure 3.5: Lossless compression (Wallace, 1991)

3.4 Image Segmentation

Image segmentation involves the process of trying to separate out a region or some regions of an image of interest. This operation is very used in medical image processing, where some regions of an image whole image are marked out from the background. The marked or highlighted region of interest is referred to as the foreground. One common and effective technique used in image segmentation is known as image thresholding.

Figure 3.6: Edge based segmentation (Saif et al., 2012)

3.4.1 Edge detection

Image processing is a very important aspect of pattern recognition and machine learning field. It offers various techniques to manipulate image data, feature extraction, image enhancement, and image segmentation. Image manipulation techniques include image sampling for up-scaling or down-scaling, conversion to gray images, black and white, etc.

Feature extraction is a process in image processing, where some characteristics or parameters that describe an image are obtained. The features of interest usually vary for different problems. Generally, these features are statistical parameters that describe some important attributes of the images.

Feature extraction operations include as edge detection, corners, points, etc. These operations are very useful in reducing the amount of irrelevant or redundant information that are contained in images. Filters are special kernels which have predefined pixel values such that it achieves the particular feature extraction of interest when applied to an image. Common filters used in feature extraction are the Sobel filter, Canny filter, Gabor filter, Hough filter, etc.

Edges are boundaries between different textures. Edge also can be defined as discontinuities in image intensity from one pixel to another. The edges for an image are always the important characteristics that offer an indication for a higher frequency. Detection of edges for an image may help for image segmentation, data compression, and also help for well matching, such as image reconstruction and so on.

- Sobel edge detection

The Sobel filter is particularly suited to edge detection in image processing; the Sobel operator reinforces edges and transitions present in the source image by performing the 2D spatial computation.

It achieves edge detection by performing a convolution of an image with the Sobel filter in the x-direction and y-direction. Sobel filters are created to have maximum response to edges running horizontally and vertically in the image; these Sobel filters are described in the figure below.

-1	0	+1
-2	0	+2
-1	0	+1

(a) G_x

+1	2	+1
0	0	0
-1	-2	-1

(b) G_y

Figure 3.7: Sobel operator

where, G_x and G_y are the Sobel filters that operate on the horizontal and vertical edges respectively. Each filter computes gradient components in either orientation (vertical or horizontal). G_x and G_y can be combined to obtain the absolute gradient component magnitudes at each point with the formula in equation 3.2.

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{3.2}$$

A much faster approximate formula for computing the absolute gradient is given in equation 3.3.

$$|G| = |G_x| + |G_y| \tag{3.3}$$

- Canny edge detection

This edge detection is an algorithm or multi-stage process that is run in order to achieve a high end or an optimal detection result. The Canny edge detector algorithm is described below.

1. Source image is smoothened out with a Gaussian filter by convolution; in this stage noise is removed from the image.
2. An edge detection filter, such as Sobel or Robert Cross, is applied to compute (highlight) regions in the image with high first derivatives (reinforce edges in the image).
3. The algorithm tracks along the edges and sets all pixels not on the ridges to zero, hence, the earlier computed edges now appear as thin lines in the image, a process referred to as non-maximal suppression.
4. The edge tracking algorithm can be considered to exhibit hysteresis in that it discourages discontinues of noisy edges into fragments during ridge tracking; it is controlled by two

threshold values, T1 and T2 (with T1>T2). Tracking starts at T1, continues in both axes till the height of the ridge reduces below T2.

The Canny edge detection algorithm is more effective, robust and gives a better outcome than the Sobel detection technique. Figure 3.8 describes the result of using Sobel and Canny detection algorithms respectively on the source image (a).

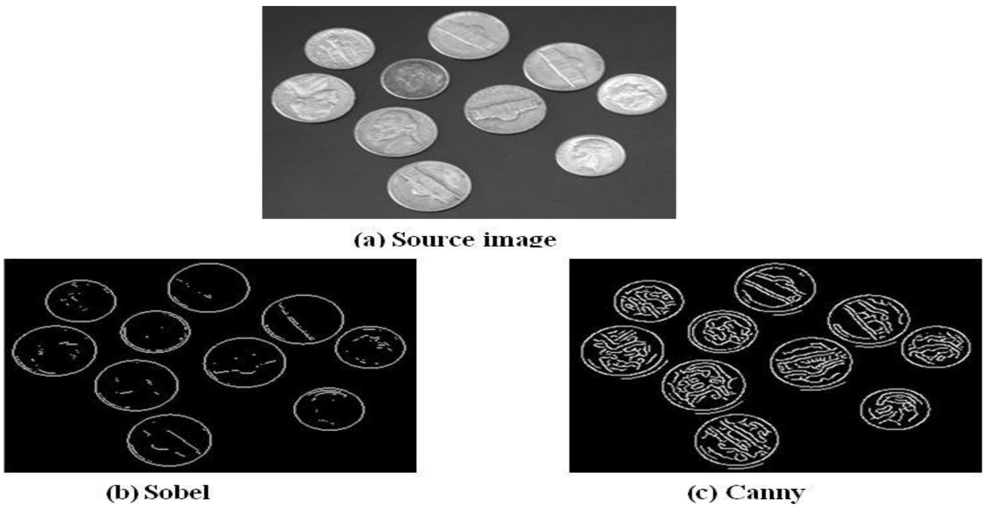


Figure 3.8: Canny and Sobel edge detections

3.5 Image Processing Applications

The field of digital image has rapidly expanded in the recent years. The usefulness of this technology is clear in many different disciplines and areas (Andrew, 2008).

The fields of image processing are:

- Robotics
- Medical imaging
- Machine vision
- Digital camera images

3.5.1 Medical image processing

Restorative imaging has been experiencing an insurgency in the previous decade with the coming of quicker, more precise, and less obtrusive gadgets. This has driven the requirement for relating programming improvement which thusly has given a noteworthy catalyst to new calculations in sign and picture transforming (Stefanescu et al., 2004).

In particular, in therapeutic imaging we have four key issues:

1. Segmentation - automated methods that create patient-specific models of relevant anatomy from images;

2. Registration - automated methods that align multiple data sets with each other;

3. Visualization - the technological environment in which image-guided procedures can be displayed;

Imaging innovation in Medicine made the specialists to see the inside parts of the body for simple determination. It likewise helped specialists to make keyhole surgeries for coming to the inside parts without truly opening excessively of the body. CT Scanner, Ultrasound and Magnetic Resonance Imaging assumed control x-beam imaging by making the specialists to take a gander at the body's subtle third measurement. With the CT Scanner, body's inside can be uncovered with straight forwardness and the unhealthy territories can be distinguished without bringing about either uneasiness or torment to the patient. X-ray grabs signals from the body's attractive particles turning to its attractive tune and with the assistance of its intense PC, changes over scanner information into uncovering pictures of inward organs. Image processing strategies produced for breaking down remote sensing information may be altered to dissect the yields of therapeutic imaging frameworks to get best preference to break down indications of the patients without any difficulty (Rao, 2004).

3.5.2 Computerized image processing requirements for medical applications

- Interfacing Analog yields of sensors for example, magnifying lens, endoscopes, ultrasound and so forth, to digitizers and thusly to Computerized Image Processing frameworks (Fan et al., 2002).

• Image upgrades.

• Changing thickness element scope of B/W images.

• Color redress in shading images.

• Manipulating of hues inside an image.

• Contour discovery.

• Area estimations of the cells of a biomedical picture.

• Display of picture line profile.

• Restoration of images.

• Smoothing of images.

• Registration of different images.

• Construction of 3-D images from 2-D images.

• Generation of negative images.

• Zooming of images.

• Pseudo shading.

• Point to point estimations.

• Getting help impact.

CHAPTER 4
ARTIFICIAL NEURAL NETWORK

In this chapter the review of artificial Neural Network is presented. The advantage and disadvantage, network models, algorithms are described. The ORC (Optical Character Recognition) and Intelligent Transportation system are presented.

4.1 What is ANN?

Artificial neural networks (ANNs) are the simple simulation of the structure and the function of the biological brain. The complex and accurate structure of the brain makes it able to do hard different simultaneous tasks using a very huge number of biological neurons connected together in grids. A first wave of interest in neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks (Krose & Smagt, 1996). At that time, Von Neumann and Turing discussed interesting aspects of statistical and robust nature of brain-like information processing. But it was only in 1950s that actual hardware implementations of such networks began to be produced (Fyfe, 1996). ANNs are used widely nowadays in different branches of science. It is used for medical purposes like in (Khashman, 1999) and (Khashman, 2000). Used for image processing for different purposes like (Khashman & Dimililer, 2007). It is also invested in power and power quality applications and active power filters (Valiviita, 1998) and (Sallam & Khafaga, 2002). In (Yuhong & Weihua, 2010) a survey on the application of the ANNs in forecasting financial market prices, financial crises, and stock prediction was presented.

The different mentioned applications of neural networks imply firstly the learning of the ANNs to do defined tasks. One of the most common methods of teaching ANNs to perform given tasks is the back propagation algorithm. It is based on a multi-stage dynamic system optimization method proposed by Arthur E. Bryson and Yu-Chi Ho in 1969 (Ho, 1969). In 1974, it was applied in the context of ANNs through the works of Paul Werbos, David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, and it became famous and led to a renaissance in the field of artificial neural networks.

4.2 Analogy to the Human Brain

The artificial neural network is an imitation of the function of the human biological brain. It's using the structure and the function of brain. The human brain is composed of billions of interconnected neurons. Each one of these neurons is said to be connected to more than 10000 neighbor neurons. The connecting lines are the dendrites and axons that connect between the (Shen & Wang, 2012). The dendrites receive the electrochemical signals from the other cells and transmit it to the body of the cell. If the signals received are powerful enough to fire the neuron; the neuron will transmit another signal through the axon to the neighbor neurons in the same way. The signal is going also to be received by the connected dendrites and can fire next neurons.

4.3 Artificial Neural Networks

Artificial neural networks are a structure that has inspired its origins from the human thinking centre or the brain. This structure has been inspired and developed to build a mechanism that can solve difficult problems in the science. Most of the structures of neural networks are similar to the biological brain in the need for training before being able to do a required task (Kaki, 2009). Similar to the principle of the human neuron, neural network computes the sum of all its inputs. If that sum is more than a determined level, the correspondent output can then be activated. Otherwise, the output is not passed to the activation function. Figure 4.1 presents the main structure of the artificial neural network where we can see the inputs and weights in addition to the summation function and the activation function. The output function is the output of the neuron in this structure. The input of the activation function is given by:

$$TP = \sum x_n \omega_n \tag{4.1}$$

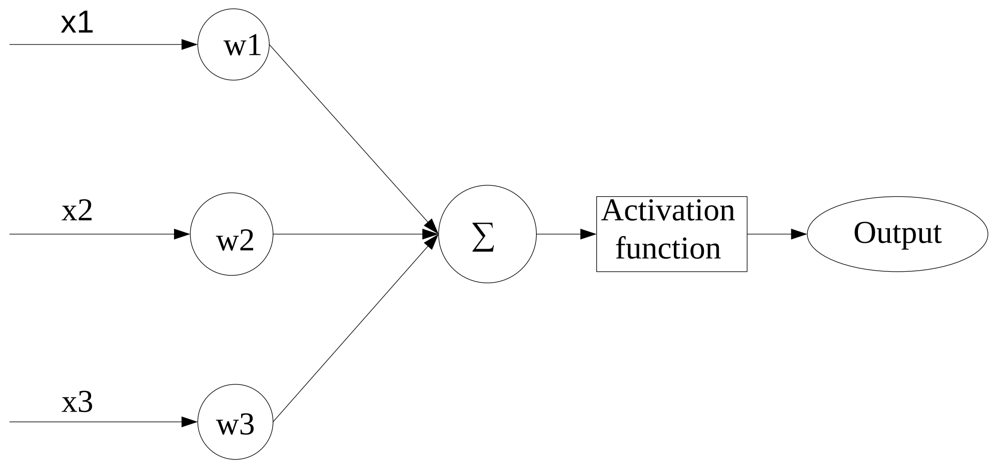


Figure 4.1: Basic structure of artificial neural network

4.3.1 Structure of ANN

The structure of ANNs consists mainly of three aspects in addition to the learning method. These aspects are the layers, weights, and activation functions. Each one of these three parts play a very important rule in the function of the ANN. The learning function is the algorithm that relates these three parts together and ensures the correct function of the network.

4.3.2 Layers

ANN is constructed by creating connections between different layers to each other. Information is being passed between the layers through the synaptic weights. In a standard structure of ANN there are three different types of layers (Mena, 2012):

1- Input layer: the input layer is the first one in a neural network. Its rule is the transmission of input information to the other layers. An input layer doesn't process the information; it can be considered as the sensors in biological system. It can also be called non processing layers.
2- Output layer: The last layer in the neural network whose output is the output of the whole network. In contrary to the input layer, the output layer is a processing layer.
3- Hidden layers: this is the main part of the network. It consists of one or more of processing layers. They are connecting the input layers to the output layers. Hidden

layers are the main processing layers where the weights are being updated continuously. Each one of the hidden layers connects between two hidden layers or one hidden and input or output layer.

Figure 4.2 presents the layers of the neural network and the connections between the layers. As shown in the figure, the inputs are fed to the input layer. The output of the input layer is fed to the hidden layers. The output obtained from the hidden layers is fed to the output layer that generates the output of the network.

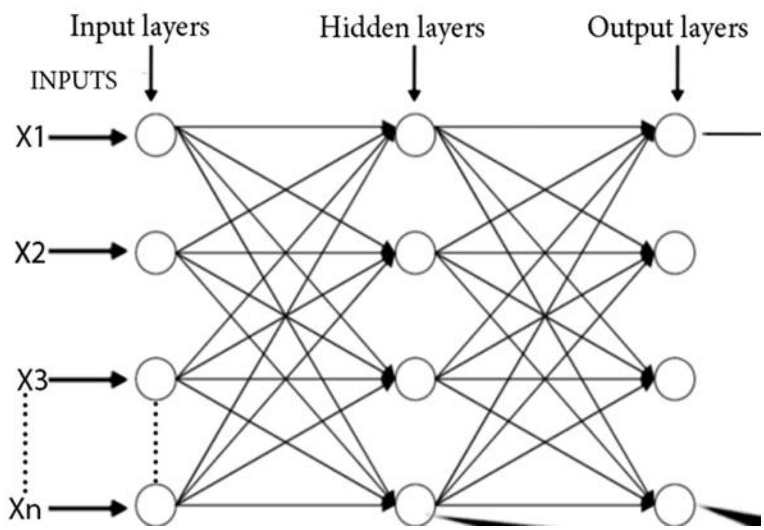


Figure 4.2: Layers structure in ANNs

4.3.3 Weights

The weights in an ANN represent the memory of that network in which all information is stocked. The values of the weights are updated continuously during the training of the network until the desired output is reached. The memory or weights are then stored to be used in future. After learning the values of these weights are used as the memory of network (Roberts, 2015).

4.3.4 Activation functions or transfer functions

When the inputs are fed to the layers through the associated weights and finding the sum of them, an activation or transfer function is used to determine whether the output is to be activated or not. Or in some activation functions, the function is used to determine how much the processed input will share in constructing the total output of the network. Activation functions are very important in neural networks because they can decide whether the input to the neuron is enough to be passed to the next layer or not (Mena, 2012). There are many types of activation functions in artificial neural networks:

4.3.4.1 Linear activation functions or ramp

In this type of the activation function, the output is varies linearly when the input is small (Yuhong and Weihua, 2010). If the input is large, the absolute output is limited by 1 as shown in figure 4.3. The function of this transfer function is defined by:

$$o(TP) = \begin{cases} -1 & TP \leq -1 \\ TP & -1 \leq TP \leq 1 \\ 1 & 1 \leq TP \end{cases} \tag{4.2}$$

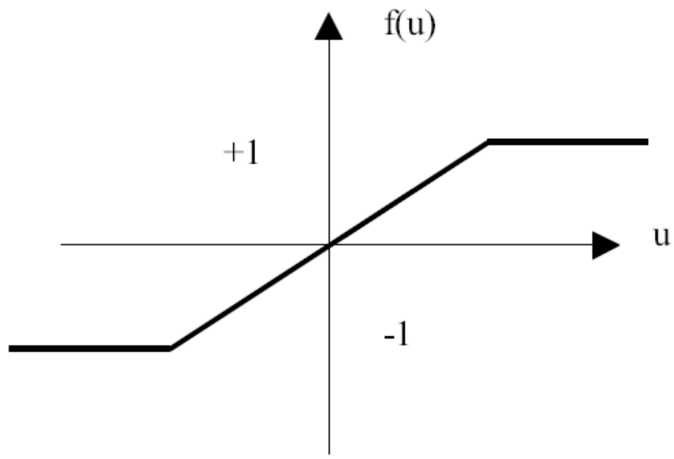


Figure 4.3: Ramp activation function

4.3.4.2 Threshold function (Hard activation function)

In the threshold function the output is zero if the summed input is less than certain value of threshold, and 1 if the summed input is greater than threshold. This way the output is oscillating between two values (Yuhong & Weihua, 2010). It can be either activated or deactivated like in figure 4.4. The function of the hard function is defined by:

$$o(TP) = \begin{cases} 0, & TP < \theta \\ 1, & TP > \theta \end{cases} \tag{4.3}$$

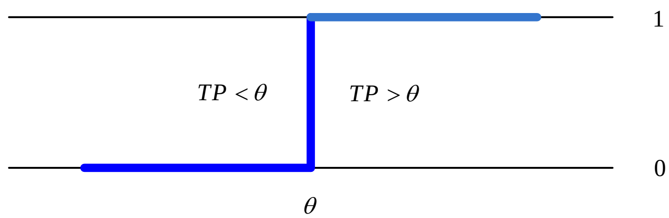


Figure 4.4: Hard activation function

4.3.4.3 Sigmoid function

This function can range between 0 and 1, but in some cases it can be useful to range it between -1 and 1. The logarithmic sigmoid and hyperbolic tangent is of the most common sigmoid functions. These two functions are the most used in the back propagation because they are differentiable. The formulas of these two functions in addition to the curves are presented in figure 4.5. The slope of the curves can be varied based on the application for which it is used (Kaki, 2009).

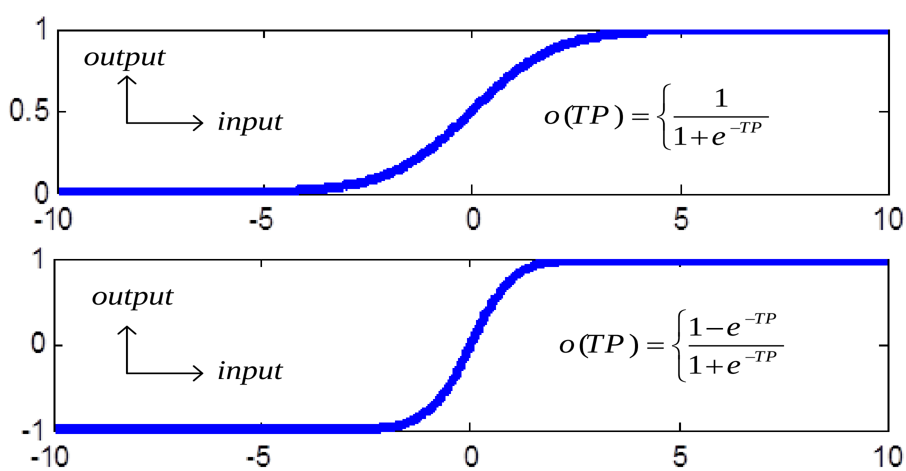


Figure 4.5: logarithmic and hyper tangential sigmoid activation functions

In the back propagation algorithms, the log-sig and tan-sig functions are the most used (Kaki, 2009). The main advantage of these two functions is the fact that they can be easily differentiated. The derivative of the logarithmic sigmoid is given by:

$$\frac{d}{dt}o(\theta) = o(\theta) * (1 - o(\theta)) \tag{4.4}$$

CHAPTER 5
SCALE INVARIANT FEATURE TRANSFORM (SIFT)

In 2004, D. Lowe, University of British Columbia, contrived another study, Scale Invariant Feature Transform (SIFT) in his paper, Distinctive Image Features from Scale-Invariant Keypoints, which separate keypoints and technique its descriptors (Lowe, 2004).

In various figurings, we saw some corner pioneers like Harris and so on. They are upset invariant, which deduces, paying little regard to the probability that the photograph is pivoted; we can locate the same corners. It is clear in light of the way that corners remain corners in turned picture furthermore. Regardless, shouldn't something be said as to scaling? A corner may not be a corner if the photograph is scaled. For example, check a crucial picture underneath. A corner in a little picture inside a little window is level when it is zoomed in the same window. So Harris corner is not scale invariant.

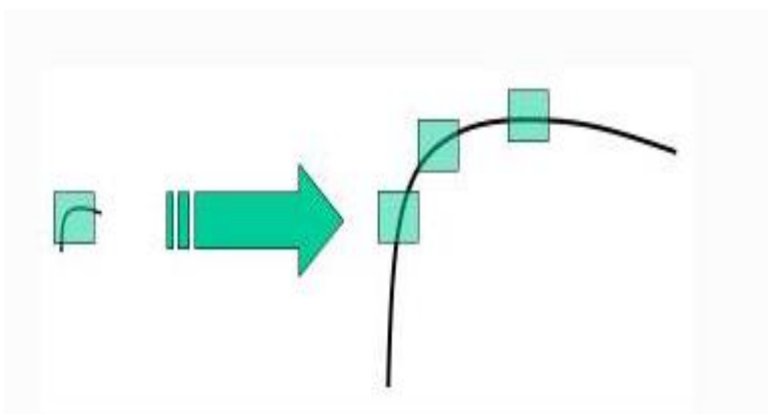


Figure 5.1: Scale-invariance (Lowe, 2004)

Four steps are involved in the SIFT algorithm. They will be discussed one by one in this chapter.

5.1 SIFT - Scale Invariant Feature Transforms

For any item there are numerous elements, intriguing point on the object that can be extricated to give a "feature" portrayal of the item. This depiction can then be utilized when endeavoring to find the article in a picture containing numerous different items. There are numerous contemplations while separating these components and how to record them. Filter image highlights give an arrangement of elements of an article that are not influenced by a hefty portion of the confusions experienced in different techniques, for example, object scaling and pivot (Lowe, 1999).

While taking into consideration an article to be perceived in a bigger picture SIFT picture includes additionally take into account objects in various pictures of the same area, taken from various positions inside the earth, to be perceived. Filter components are likewise exceptionally strong to the impacts of "clamor" in the picture.

The SIFT approach, for picture highlight era, takes a picture and changes it into a "vast accumulation of neighborhood highlight vectors (Lowe, 2004). Each of these element vectors is invariant to any scaling, pivot or interpretation of the image. This methodology offers numerous components with neuron reactions in primate vision. To help the extraction of these components the SIFT calculation applies a 4 phase separating approach.

5.1.1 Scale-space extrema detection

From the image above, plainly that the same window was utilized to recognize keypoints with different scale. It supports of little corner. In any case, to recognize greater corners we require greater windows. For this, scale-space separating is utilized. In it, Laplacian of Gaussian is found for the picture with different \sigma values. LoG goes about as a blob pointer which sees blobs in different sizes in perspective of progression in \sigma. Basically, \sigma goes about as a scaling parameter. For eg, in the above picture, gaussian part with low \sigma gives high respect for little corner while guassian bit with high \sigma fits well for more prominent corner. Thusly, we can discover the range maxima over the scale and space which gives us a rundown of (x,y,\sigma) qualities which surmises there is a potential keypoint at (x,y) at \sigma scale (Lowe, 2001).

By the by, this LoG is to some degree amazing, so SIFT tally utilizes Difference of Gaussians which is an estimation of LoG. Intricacy of Gaussian is gotten as the capability of Gaussian blurring of a picture with two different \sigma, let it be \sigma and k\sigma. This technique is capable for various octaves of the picture in Gaussian Pyramid. It is tended to in underneath picture:

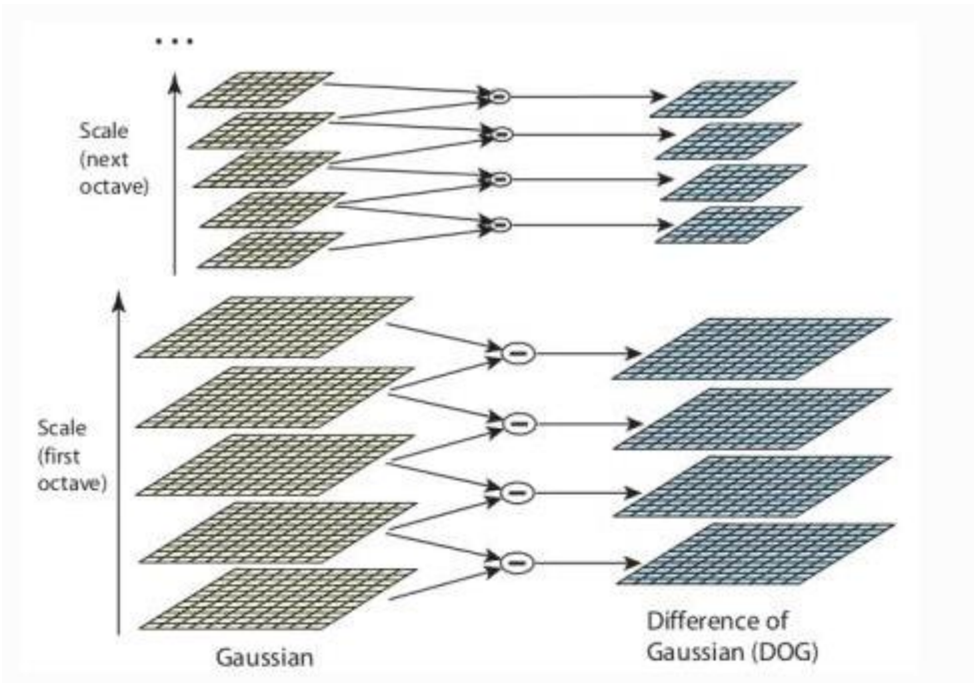


Figure 5.2: Gaussian pyramids (Lowe, 2004)

When this DoG are found, images are scanned for nearby extrema over scale and space. For eg, one pixel in an image is contrasted and its 8 neighbors and in addition 9 pixels in next scale and 9 pixels in past scales. On the off chance that it is a neighborhood extrema, it is a potential keypoint. It essentially implies that keypoint is best spoken to in that scale. It is appeared in underneath image:

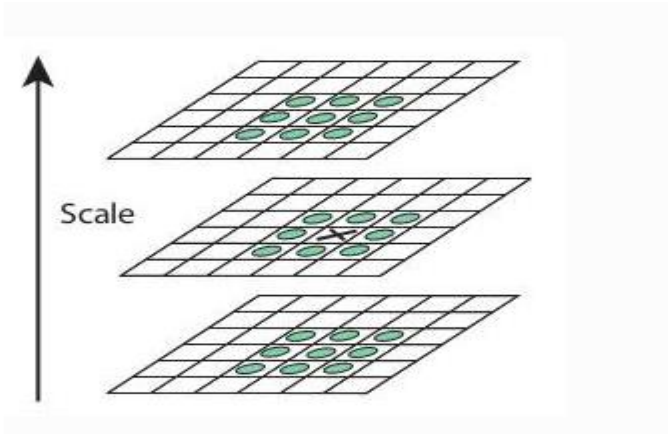


Figure 5.3: Scale-invariance (Lowe, 2004)

As to parameters, the work gives some accurate data which can be compressed as, number of octaves = 4, number of scale levels = 5, starting \sigma=1.6, k=\sqrt{2} and so forth as ideal qualities (Lowe, 2001).

The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{5.1}$$

Where * is the convolution administrator, G(x, y, σ) is a variable-scale Gaussian and I(x, y) is the info image.

Different systems can then be utilized to distinguish stable keypoint areas in the scale-space. Difference of Gaussians is one such system, finding scale-space extrema, D(x, y, σ) by figuring the difference between two images, one with scale k times the other. D(x, y, σ) is then given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{5.2}$$

To detect the nearby maxima and minima of D(x, y, σ) every point is contrasted and its 8 neighbors at the same scale, and its 9 neighbors here and there one scale. On the off chance that this worth is the base or most extreme of every one of these focuses then this point is an extrema.

5.1.2 Keypoint localization

This stage endeavors to take out more focuses from the rundown of keypoints by finding those that have low differentiation or are ineffectively confined on an edge. This is accomplished by figuring the Laplacian, The location of extremum, z, is given by:

$$\boldsymbol{z} = -\frac{\partial^2 \boldsymbol{D}^{-1}}{\partial \boldsymbol{x}^2} \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{x}} \tag{5.3}$$

In the event that the capacity esteem at z is underneath edge esteem then this point is rejected. This expels extrema with low complexity. To dispense with extrema in view of poor localisation it is noticed that in these cases there is a huge standard bend over the edge however a little ebb and flow in the opposite course in the difference of Gaussian capacity. On the off chance that this difference is beneath the proportion of biggest to littlest eigenvector, from the 2x2 Hessian matrix at the area and size of the keypoint, the keypoint is rejected (Lowe, 2001).

5.1.3 Orientation assignment

In short an acquaintance is relegated with each keypoint to accomplish invariance to picture change. A neigbourhood is taken around the keypoint range contingent on the scale, and the inclination size and course is enlisted in that area. A presentation histogram with 36 storehouses covering 360 degrees is made. (It is weighted by point immensity and gaussian-weighted meandering window with \sigma equivalent to 1.5 times the range of keypoint. The most lifted top in the histogram is taken and any crest above 80% of it is in like way considered to figure the presentation. It makes keypoints with same area and scale, yet unmistakable heading. It adds to security of sorting out (Rayella & Hanson, 2001).

5.1.4 Keypoint descriptor

The nearby slope information, utilized above, is additionally used to make keypoint descriptors. The slope data is pivoted to line up with the orientation of the keypoint and afterward weighted by a Gaussian with change of 1.5 * keypoint scale. This information is then used to make an arrangement of histograms over a window fixated on the keypoint.

Keypoint descriptors commonly utilizes an arrangement of 16 histograms, adjusted in a 4x4 matrix, each with 8 orientation canisters, one for each of the primary compass bearings and one for each of the mid-purposes of these headings. These outcomes in a component vector containing 128 components (Yanushkevich, et al., 2008).

5.1.5 Keypoint matching

Keypoints within two images are coordinated by distinguishing their closest neighbors. Be that as it may, sometimes, the second nearest match might be exceptionally close to the first. It might happen because of commotion or some different reasons. All things considered, proportion of nearest separation to second-nearest separation is taken. In the event that it is more noteworthy than 0.8. It eliminates around 90% of false matches while disposes of just 5% right matches, according to the paper (Lowe, 2004).

5.2 Summary

This chapter discussed the Scale Invariant Feature Transform (SIFT) that recently becomes an effective technique for the task of face recognition. The chapter presented an introduction in addition to a detailed explanation of the SIFT concept and working principles.

CHAPTER 6
THE SYSTEM DESIGN AND PERFORMANCE

6.1 Face Recognition and SIFT in Image Processing

Face Recognition (FR) is an exploration range spreading over a few trains, for example, image processing, pattern recognition, computer vision and neural system. There are numerous uses of FR as appeared in Table 1. These applications range from coordinating of photos to ongoing coordinating of reconnaissance video. Contingent upon the particular application, FR has diverse level of trouble and requires extensive variety of strategies. In 1995, a survey paper by (Chellappa et al., 2010) gives a thorough review of FR around then. Amid the previous couple of years, FR is still under quick development.

The points of the proposed framework is to utilize the Scale Invariant Feature Transform as highlight descriptor and extractor of a smart face recognition framework in view of backpropagation neural system. Filter descriptor contained a technique for identifying interest focuses from a dark level image at which insights of nearby slope bearings of image powers were aggregated to give an outlining portrayal of the neighborhood image structures in a neighborhood around every interest point, with the aim that this descriptor ought to be utilized for coordinating relating interest focuses between various images.

6.2 The Proposed Methodology

The proposed system is a face recognition intelligent system based on SIFT algorithm for the feature extraction and backpropagation neural network. The purpose of this research is to evaluate the effectiveness of a backpropagation neural network in recognizing different face and to compare the obtained results with those in the literature review. The developed framework consists of two main phases which are the processing phase and the classification phase in which the image is classified as different faces. In the image processing phase the face images are pre-processed using many techniques such as conversion to grayscale and filtering using median filter. Then the most significant technique takes place which is the feature extraction using SIFT. These techniques are done in order to enhance the quality of images and to extract the important features in such a way to take only the important face's features and ignoring the other features and parts of the image. At the end of this phase, the images are fed to a backpropagation neural network in which they are classified as different faces for different individuals.

The two main phases of the proposed face recognition system are illustrated in Figure 6.1

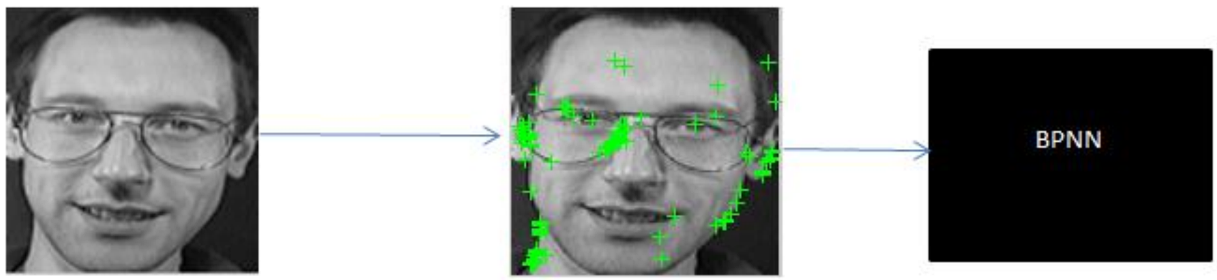


Figure 6.1: Phases of the developed face recognition system

These following are the image processing techniques and the classification methods used in our proposed system for the intelligent face recognition of individuals using SIFT.

1. Read RGB images
2. Image size rescaling to 100*100 pixels for the purpose of faster processing
3. Scale Invariant Feature Transform: SIFT
   - Scale-space extrema detection
   - Keypoint localization
   - Orientation assignment
   - Keypoint descriptor
   - Keypoint matching
4. Feed the extracted features into a backpropagation neural network
5. Train the neural network
6. Test the neural network

The analysis and processing of the face images take place first in the system so that a free-noise image is extracted from the original image. The later stages are the feature extraction and neural classification phases in which the features is extracted using the Scale Invariant Feature Transform: SIFT. Once the features are detected and extracted, they are fed into a backpropagation neural network respectively with their targets.

Figure 6.2 represents a flowchart that illustrates our proposed system for the face recognition using SIFT. Figure 6.3 shows a face image from our database that undergoes all the system processes in order finally to be segmented.

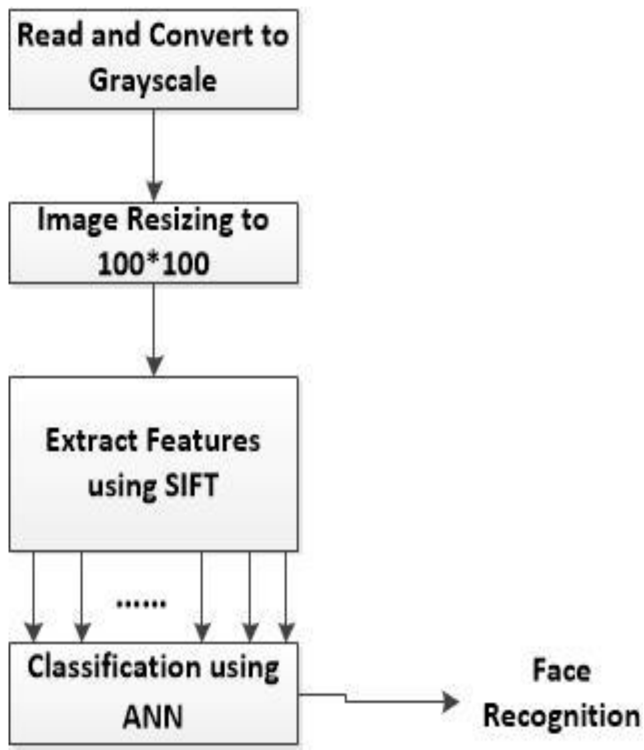


Figure 6.2: Flowchart of the developed framework

Figure 6.3: One face image processed using the developed image processing system

6.3 Dataset

The images were collected from the benchmark database: **AT&T database.** This database contains 400 images for 40 subjects, with 40 images/person. The images contain different facial expressions and illumination conditions for each subject. The image size is 256*256 pixels, 128 SIFT features are extracted for each image.

The images were all resized to 100*100 pixels for fast processing purposes. The total number of images used for the designed system is 200 images. Among them, 100 are for training and 100 for testing phase. The 200 images are divided into 20 individuals; each one of them has 10 different facial expressions. For each individual 5 expressions were used for training while the other 5 expressions were used for testing purposes. Table 6 shows the number of face images in the database. Figure 6.4 illustrates some of the images found in the AT&T database.

Table 6: Total number of images

Number of individuals	Nb. of expression per individual	Nb. Of poses per expression	Total
20	10	1	200

In order to improve the effectiveness of the network, some images are rotated in different angles more than one time so that the network has all the required properties such as rotation-invariance and Scale invariance aims to make the intelligent system more robust in determining the image of faces that can be placed at different angles (Khashman 2012).

Moreover, the purpose of getting different facial expression images to use the other facial images for testing phase in order to evaluate the effectiveness of the designed face recognition system.

Figure 6.4: Sample of the database images

6.4 The System Design Process

SIFT algorithm is a technique that is used to extract some unique, scaled, and invariant features that distinguish between different faces. Thus, using these features for distinguishing faces would come up with an efficient system that might be capable of recognizing faces robustly regardless of the face image scale or orientation. Thus, the SIFT algorithm was used in designing the presented face recognition system as a scale –invariant feature extractor. These features represent the each face since they differ from each image to another. Therefore, they are used as inputs for the neural classifier that would learn and generalize (recognize) faces later on in the testing phase.

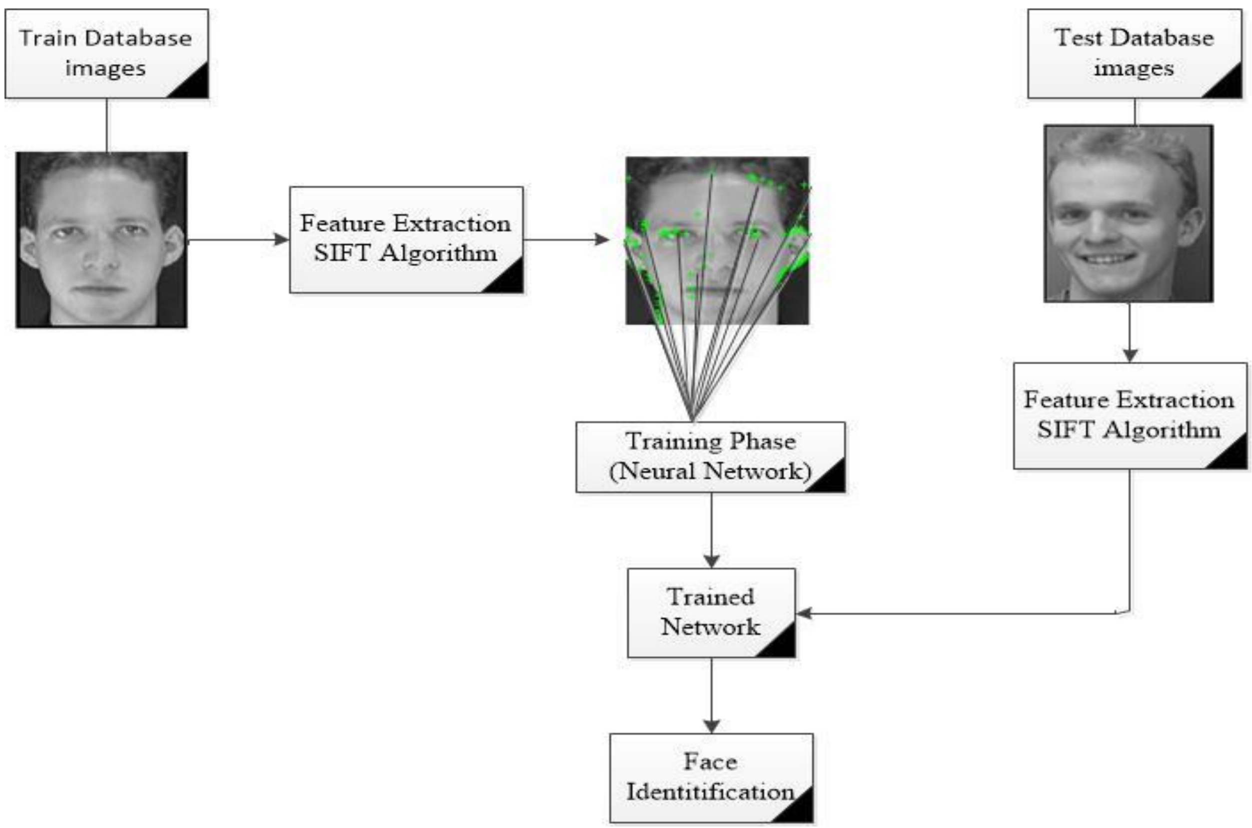


Figure 6.5: System Flowchart

The Figure above shows the flowchart of the proposed intelligent face recognition system using SIFT algorithm. The images are used to train the backpropagation network through their features which are extracted using SIFT algorithm. After training and convergence, the testing images are used then for testing the neural network after they pass through the feature extraction phase using SIFT.

6.5.1 Features Extraction

As mentioned above, the faces were not directly fed to the neural network. However, an algorithm was used to extract some features that can represent the whole face and be unique and rotation invariant. This algorithm is called Scale Invariant Feature Transform.

Scale Invariant Feature Transform (SIFT) is an image descriptor used to detect for image-based matching and recognition developed by David Lowe (Lowe, 2004).

The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors" (Lowe, 1999). Each of these feature vectors is invariant to any scaling, rotation or translation of the image.

To aid the extraction of these features the SIFT algorithm applies a 4 stage filtering approach:

- Scale-Space Extrema Detection

This phase of the separating endeavors to recognize those areas and scales that are identifiable from various perspectives of the same article.

- Keypoint Localistaion

This stage endeavors to kill more focuses from the rundown of keypoints by finding those that have low differentiation or are ineffectively limited on an edge. This is accomplished by ascertaining the Laplacian.

- Orientation Assignment

This progression plans to appoint a steady introduction to the keypoints taking into account neighborhood image properties. The keypoint descriptor, depicted underneath, can then be spoken to with respect to this introduction, accomplishing invariance to rotation.

- Keypoint Descriptor

The neighborhood angle information, utilized above, is additionally used to make keypoint descriptors. The slope data is rotated to line up with the introduction of the keypoint and after that weighted by a Gaussian with fluctuation of 1.5 * keypoint scale.

The figure below shows the steps of the SIFT approach when applying in the proposed system in order to detect the keypoints of image.

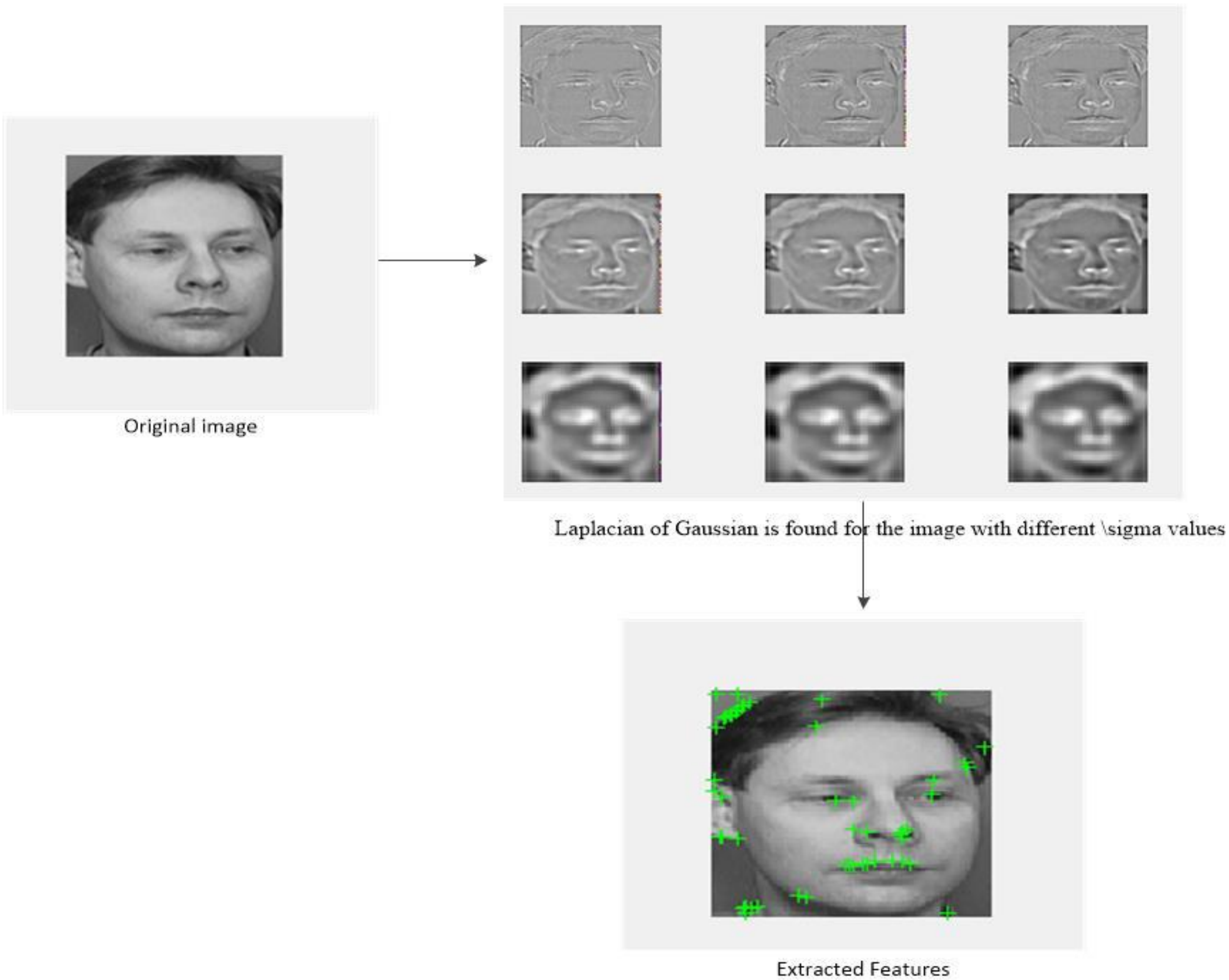
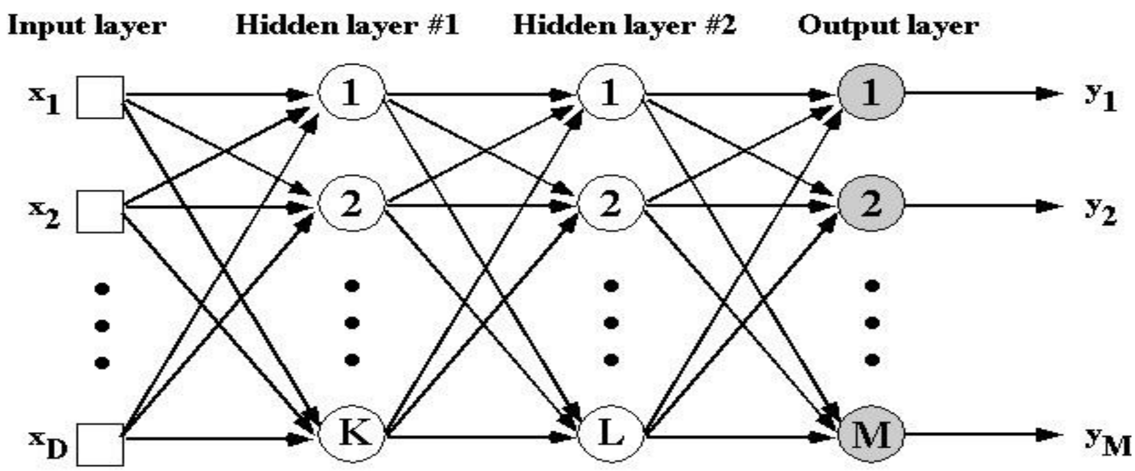


Figure 6.6: SIFT approach applied on the proposed system images

6.6 Training of the Neural Network System

6.6.1 Backpropagation neural network

It is important that while a single neuron is capable of learning, there are some complex functions that cannot be satisfactorily learned by a single neuron. This particular set of problems are referred to as linearly non-separable problems; that is one single straight line cannot be used to satisfactorily partition decision boundaries. Note that the same problem exists in a model with many neurons of only one layer. It is has been shown that the solution to such problems relies on neural models of more than one layer. i.e. multilayer networks. Fortunately, one of the most popular multilayer network models, backpropagation neural network (BPNN), is employed within this work for modeling climate data and for performing forecast. Also, in many literatures, it is not uncommon to find backpropagation neural network referred to as multilayer perceptron (MLP). Furthermore, the backpropagation neural network relies on learning scheme referred to as supervised learning for learning tasks.



6.7: Backproagation Neural Network BPNN (Zhao, 2000)

The supervised learning scheme is a situation where a model is supplied inputs and corresponding desired outputs (or targets). The backpropagation neural network is basically a stacked of artificial neurons as layers (James et al., 2005). Backpropagation neural networks have at least three layers, which are the input, hidden and output layers.

The input layer is where input (independent) variables are supplied to the network, the hidden layer is primarily where the abstract features (associations) between the independent and dependent variables are extracted (or learned) and the output layer is where the computed dependent and target dependent are used to obtain network error for iteratively updating the parameters of the network. Note that backpropagation neural network can have more than one hidden layer; however, one hidden layer is sufficient for learning most tasks. The backpropation neural network is shown in Figure 6.7.

6.6.2 Neural network training

During this training phase, the face images are used for training the neural network in order to have the capability to recognize different scaled, noisy, and rotated images later on after convergence. We used the backpropagation algorithm as a learning method due to its simplicity and the sufficient number of images. The images were collected from the AT&T database. The database contains 200 images for 20 individuals; each has 10 different facial expressions. The system was trained on 100 images; for 20 different individuals. We used 5 different facial expressions for each individual. Therefore, the total number of images used for this phase is 100.

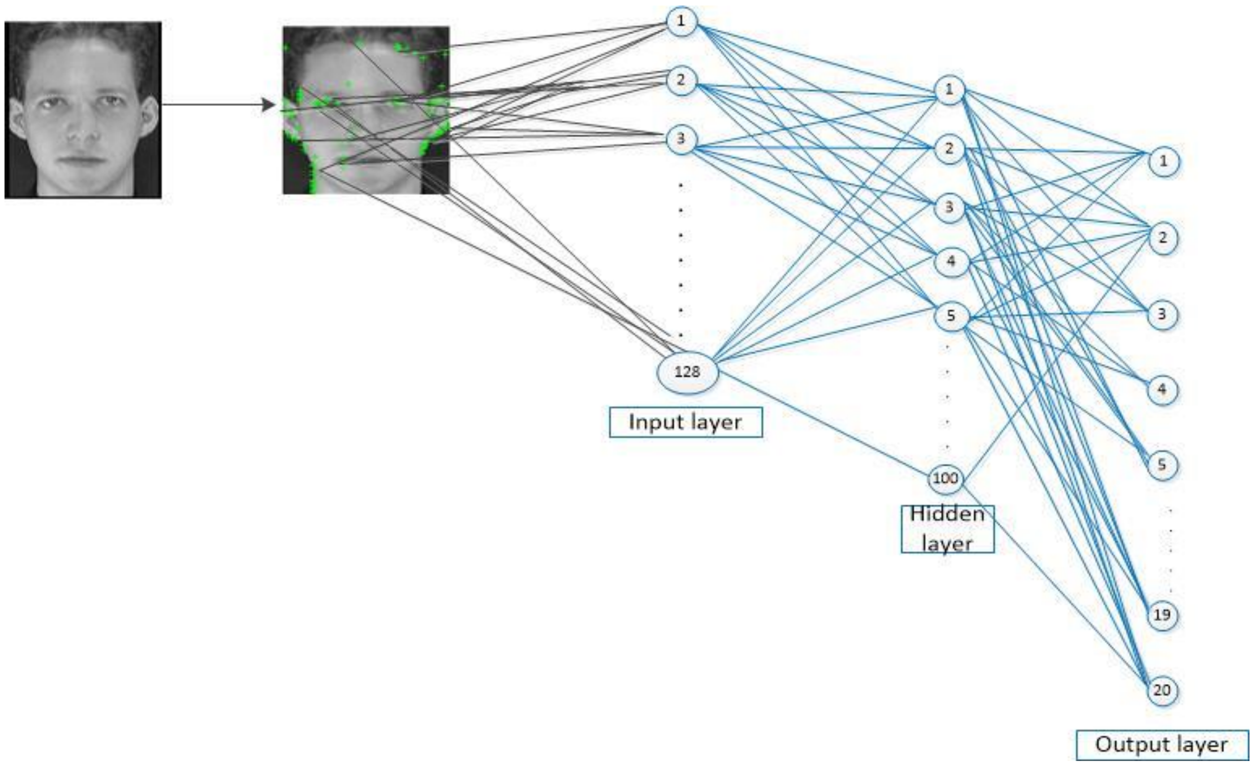


Figure 6.8: Neural network of the developed face recognition system with 100 hidden neurons

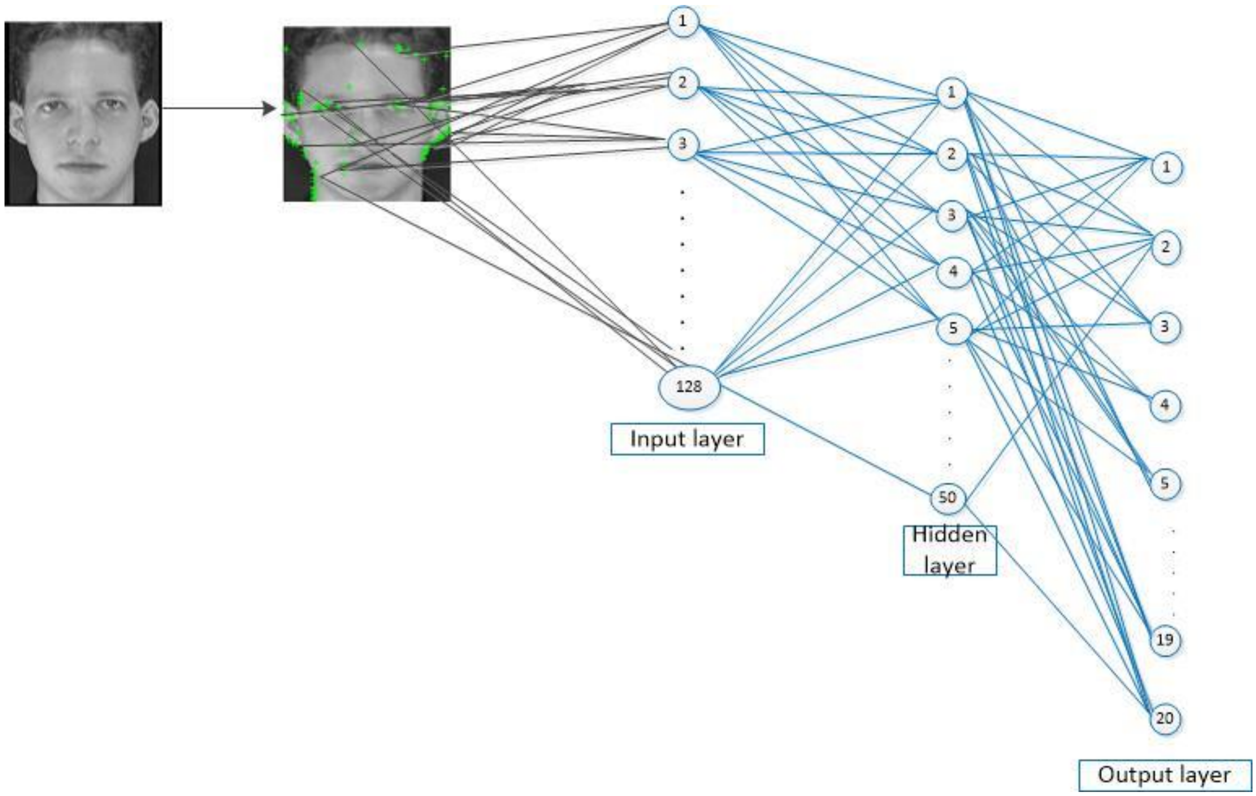


Figure 6.9: Neural network of the developed face recognition system with 50 hidden neurons

A backpropagation neural technique was used for leaning the extracted features from the SIFT algorithm. The network consists of 128 neurons since extracted descriptors are 128. The hidden layer consists of 100 neurons; this was decided after training the network for many runs. The suitable number of hidden neurons, h, is determined heuristically alongside other training parameters during training of the network. However, the output layer has 20 neurons since we have 20 different individuals.

Figure 6.9 shows the neural network topology of our proposed intelligent face recognition system based SIFT algorithm.

Table 7 presents the input parameters setting of the system. It shows all the parameters used when training the network. The maximum iteration number was set to 5000 epochs with a learning rate of 0.47, a momentum rate of 0.85 and a minimum error of 0.001. The network ran for 35 mins during the processing and training; this is due to the feature extraction which takes long time to execute its 4 steps.

Table 7: Training parameters of the network

Parameters	Value BPNN1	Value BPNN2
Number of neurons in input layer	128	128
Number of neurons in output layer	20	20
Number of neurons in hidden layer	100	50
Maximum Iteration number	5000	5000
Learning rate	0.47	0.47
Momentum rate	0.85	0.85
Error	0.001	0.001
Activation Function	Sigmoid	Sigmoid
Processing time including training	35 mins	30 mins

The type of activation function used for a neuron depends on the particular task; that is the range of expected (desired) output of the neuron. Common types of activation functions include threshold, linear, piece-wise, Logistic-Sigmoid (Log-Sigm), etc. Figure 6.10 shows the different types of activation functions.

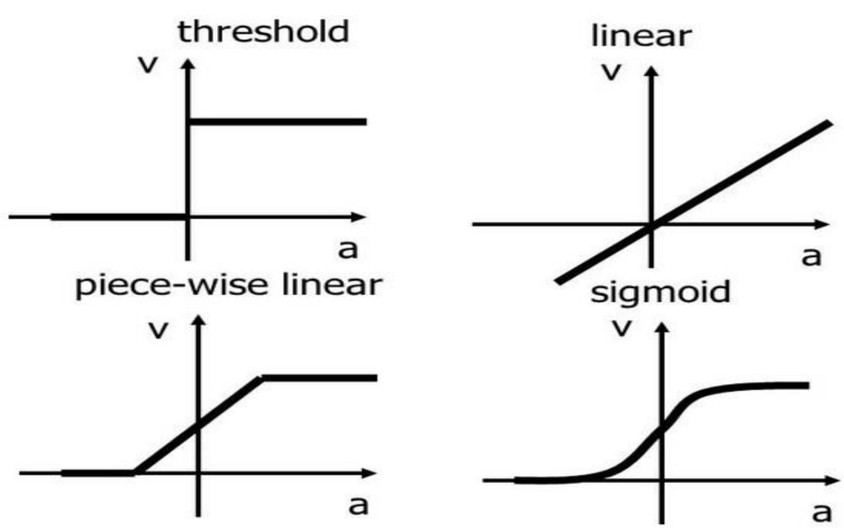


Figure 6.10: Types of activation functions

In this work, we use the sigmoid activation function which allows the learning of non-linear functions and squashes neurons output into the range 0 to 1. The sigmoid function is given in Equation 6.1.

$$v = \frac{1}{1+e^{-a}} \tag{6.1}$$

Where, a is the input to the sigmoid activation and v is the output of sigmoid function.

The following is the training results of the two sets (learning curve) for the backpropagation neural network. It can be seen that the error is decreasing while the number of iterations increases until a minimum square error of 0.0120 was obtained at epoch 4950.

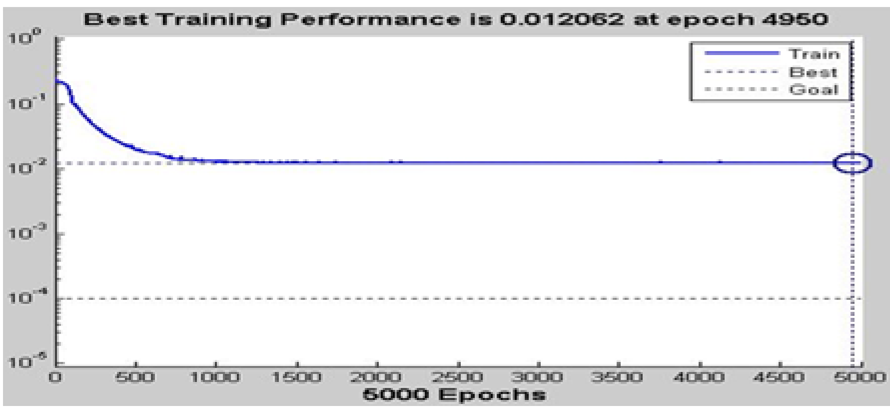


Figure 6.11: Error versus iterations variations

The training of the network resulted in 99% as a classification rate during the training phase. This means that the network leant the features of the images and only 1 image was not classified correctly.

6.7 Trained Network Performance

This intelligent face recognition system based SIFT was tested by 100 images. The 100 images are for the same individuals used in the training phase; however, these are of different facial expressions. These images have also different scales, translation and different rotation degrees.

This is done for the purpose of testing the robustness and the rotation-invariance of our face recognition system. The result of both testing and training phases is included in the following Table 8.

Table 8: The total recognition rate

Total number of images	Image sets	Number of images	Recognition Rate of BPNN1	Recognition Rate of BPNN2
	Training set	100	99%	99%
200	Testing set	100	83%	80%
	Both sets	200	91%	89%

The Table 8 above shows the recognition rate obtained in both training and testing phases of both the developed system. It also shows the number of images used in each set. The overall recognition rate obtained s 91% for BPNN1 and 89% for BPNN2.

It can be seen that the network trained well however the generalization of it was not as good as learning. However, this result is good comparing to those in the state of art.

The Table 9 below shows the Recognition rate obtained for different input parameters values. It is remarkable that the increase of the number of hidden neurons results in a better and higher Recognition rate.

Table 9: Different recognition rate for different input parameters

Learning rate	Momentum rate	Hidden neurons BPNN1	Hidden neurons BPNN2	Epochs	Recognition rate of BPNN1	Recognition rate of BPNN2
0.04	0.3	20	10	5000	81 %	80 %
0.05	0.4	40	60	5000	83 %	82.5 %
0.02	0.66	70	60	5000	89 %	89 %
0.47	0.85	100	50	5000	91 %	89 %

6.8 Results Discussion

In this thesis, an intelligent face recognition system was developed. The system is based on both image processing and neural network classification. As a processing phase, the images undergo some image enhancement techniques for the purpose of enhancing the quality of images and reducing the processing time. The Scale Invariant Feature Transform was used in designing this system due to its effectiveness in feature extraction. 128 features were extracted from each face image and fed into a neural network for neural classification. A big and enough number of images of 20 individuals were used for training the developed system. Different facial expressions of same individual were used when training. This employs a robustness of the recognition system. Different images of the same individuals but with different facial expressions were used for testing the intelligent system.

After convergence, the network was finally able to effectively recognize different faces with various facial expressions through the extracted scale-invariant features.

One of the challenges that we faced when training the system is the weights initialization. Since artificial neural network weights are usually randomly initialized at the start of training, it therefore follows that trained BPNN is not always guaranteed to converge to the global minimum or good local minima. This means that network may not achieve a minimum square error or high recognition rate during this phase. Therefore, the network has to be trained for 2 to 3 runs until the weights are correctly updated to guarantee a good network conversion or learning rate of percentage higher than 90%.

Another challenge was faced is that the network may be poor in generalization phase. In other words, the network may converge to a good local minima or achieve high recognition rate; however, it the recognition rate in the testing phase may be very low.

Many and various reasons can be the cause of this generalization problem. The network learining parameters such as momentum rate, learning coefficient, and number of hidden neurons might be a reason of this problem. Thus, changing them may change the network training recognition rate which may improve the rate in the testing phase.

Another reason is that the network may go into the overfitting problem during training. This means that the networks started to memorize the features; not learn them. This is due the high specified number of iterations set during the training of the network. Hence, the network achieved a minimum square error at a specific epoch number however; it keeps training until it goes into overfitting. This causes a weak generalization capability when testing the network since it memorized the features.

Thus, the training of the network should be stopped once a minimum square error is achieved. To do this, the training data images have to be divided into training and validation data, so that the network can check its generalization capability while training. This assures that the network will stop training once a minimum square error is achieved and that network will not go into overfitting or memorizing during training.

Nevertheless, our training data were not divided into training and validation images. Therefore, the solution for the overfitting problem was to reduce the number of iterations or epochs to a number that is exactly after the minimum square error is achieved.

Finally, solving these challenges contributed effectively to obtain a good and robust neural network that is capable of recognizing faces of different facial expressions but with a bit long processing time. This is due to the use of SIFT algorithm which generally takes long time to perform its 4 steps that lead to the extraction of the features of the face's image.

6.9 Results Comparison

As discussed previously in the literature review section, most of the researches were conducted for the purpose of recognizing faces using intelligent classifiers. Most of these researches included an image processing phase where the images are enhanced, filtered, and the faces features are extracted using different features extraction techniques. Our proposed work suggested to use the SIFT algorithm for the extraction of features from the face image in addition to the backpropagation neural network that learns these extracted features and use them later for generalization.

Lichun Zhang in (Zhang et al., 2008) investigated the use of SIFT algorithm and SVM classifier for the face recognition application. The authors used the SIFT for the features extraction of the faces and they used different database to train and test their system. Moreover, the SVM was used as an intelligent classifier to recognize the faces. One of these databases that was used in their work is the one that was used in our work AT&T. their proposed face recognition using SIFT and SVM showed a robust performance in recognizing the faces of different databases (89%).

In another work proposed in (Kisku et al., 2010); the face recognition was developed using SIFT under Multiple Paradigms of Graph Similarity Constraints. The authors have discussed the complete graph that makes is used with the invariant SIFT features. The method is produced with the three chart coordinating limitations, in particular Gallery Image based Match Constraint, Reduced Point based Match Constraint and Regular Grid based Match Constraint. The revolution, scale and fractional light invariant SIFT elements are then removed from the standardized face pictures. At long last, the diagram based topology is connected for coordinating two face pictures. The performance of this work provided a robust recognition of the faces with a high recognition rate of 91%.

Another work of face recognition using SIFT algorithm was proposed by Mohamed Aly in (Aly, 2006). The authors extracted the face features using SIFT then those features were fed into an intelligent classifier such as Nearest Neighbor in order to compare the results of the different database images. The author concluded that the accuracy of the system was better when using the AT&T database (91.7%) than using the other database: eigenface (72.1%).

The Table 10 below shows the results comparison of our proposed system with some other systems that used same databases but different classifiers. Note that all these compared researches used SIFT to extract features. It can be seen that the developed system performs well in the generalization phase since its accuracy is either equal or higher than the other proposed researches.

Table 10: Results comparison

Paper Title	Authors	Methods used	Recognition Rate
Face Recognition Using Scale Invariant Feature Transform and Support Vector Machine	Lichun Zhang et al.	SIFT and SVM	89 %
Face Recognition using SIFT under Multiple Paradigms of Graph Similarity Constraints	Dakshina Ranjan Kisku et al.,	SIFT and complete graph topology	91%
Face Recognition using SIFT Features	Mohamed Aly	SIFT and Nearest neighbor	91.7%
Proposed Face Recognition system	Mohamed Asagher and Kamil Dimililer	SIFT and Backpropagation neural network	91%

6.10 Conclusion

In conclusion, it can be stated that the use of backpropagation neural network combined with SIFT algorithm as a feature extractor was investigated in this study. The motivation of this work was evaluate the effectiveness of a backpropagation neural network based SIFT algorithm in recognizing different faces and compare the obtained results with those in the literature review. The developed systems comprises of two main phases: the processing phase and the classification phase where the images are recognized. In the first phase the face images are pre-processed using many techniques such as filtering using median filter and adjustment which results in free-noise images. Then the features of faces are extracted using SIFT algorithm. These image processing techniques are used in order to enhance the quality of images and to extract the important features in such a way to take only the important face's features and ignoring the other features and parts of the image. The next phase is to feed those extracted features into a backpropagation neural network that learns them using gradient descent learning algorithm in order to be classified as different faces for different individuals.

The images were collected from the benchmark database: AT&T database which is a database contains 400 images for 40 subjects, with 40 images/person. The images contain different facial expressions and illumination conditions for each subject which was an advantage for us to make the system more efficient and robust. The image size is 256*256 pixels, 128 SIFT features are extracted for each image.

The images were all resized to 100*100 pixels for fast processing purposes. The total number of images used for the designed system is 200 images. Among them, 100 are for training and 100 for testing phase. The 200 images are divided into 20 individuals; each one of them has 10 different facial expressions. For each individual 5 expressions were used for training while the other 5 expressions were used for testing purposes. The developed system is a robust system since it was trained to recognize faces regardless of many factors that can affect faces such as illumination, shifting, and difference in facial expressions. Thus, the system was able to recognize the individual's face with different facial expressions i.e. smiling, disgusting, eyes closed etc…

Finally, it can be stated that the experimental analysis of the proposed face recognition system showed a great efficiency and an outperforming rate over the state of art studies.

REFERENCES

Aly, M. (2006). Face recognition using SIFT features. *CNS/Bi/EE report*, *186*.

Belhumeur, P. N., & Kriegman, D. J. (1998). What is the set of images of an object under all possible illumination conditions?. *International Journal of Computer Vision*, *28*(3), 245-260.

Beumier, C., & Acheroy, M. (2000). Automatic 3D face authentication. *Image and Vision Computing*, *78*(4), 10-21.

Fan, Y., Jiang, T., & Evans, D. J. (2002). Volumetric segmentation of brain images using parallel genetic algorithms. *IEEE transactions on medical imaging*, *21*(8), 904-909.

Ghasabi, H., & Mahdavi, N. (2006). An efficient simplified neural network for solving linear and quadratic programming problems. *Applied Mathematics and Computation*, *175*(1), 452-464.

Gonzalez R.C., Woods R.E. (2001) Digital Image Processing. New York: CRC Presss.

Haider, A. M., & Kaneko, T. (2001). Automated 3D–2D projective registration of human facial images using edge features. *International journal of pattern recognition and artificial intelligence*, *15*(08), 1263-1276.

Ji, Q. (2002). 3D face pose estimation and tracking from a monocular camera. *Image and vision computing*, *20*(7), 499-511.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *In Computer vision, 1999. In The proceedings of the seventh IEEE international conference on* (pp. 1150-1157). New York University.

Lowe, D. G. (2001). Local feature view clustering for 3D object recognition. *Computer Vision and Pattern Recognition*, *193*(1-2), 129-135.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.

Marzban, C., & Viswanathan, R. (1994). Stochastic neural networks with the weighted Hebb rule. *Physics Letters A, 191*(1-2), 127-133.

Price, K., Raj, R. (2011). Picture Segmentation Using a Recursive Region Splitting Method. *Computer Graphics and Image Processing International Journal of Computer Vision*, *8*(3), 6-9.

Rao, K. M. M., & Rao, V. D. P. (2006). Medical image processing. *In Proceeding of Workshop on Medical Image Processing and Applications* (pp.605-630). New York University.

Sonka, M., Hlavac, V., & Boyle, R. (2014). *Image processing, analysis, and machine vision*. Cengage Learning.

Stefanescu, R., Pennec, X., & Ayache, N. (2004). Grid powered nonlinear image registration with locally adaptive regularization. *Medical image analysis*, *8*(3), 325-342.

Turk, M. A., & Pentland, A. P. (1991). Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 5*(1), 45-51.

Villegas, O. O. V., Balderrama, D. G., Domínguez, H. D. J. O., & Sánchez, V. G. C. (2009). License plate recognition using a novel fuzzy multilayer neural network. *International journal of computers, 3*(1), 31-40.

Warfield, S. K., Jolesz, F. A., & Kikinis, R. (1998). A high performance computing approach to the registration of medical imaging data. *Parallel Computing*, *24*(9), 1345-1368.

Zhao, W., & Chellappa, R. (2000). 3D model Enhanced Face Recognition. *In Proceedings of the International Conference on Image Processing* (pp. 50 -53). University of Hong Kong.

Zhao, W., & Chellappa, R. (1999). *Robust image based 3 D face recognition* (Doctoral dissertation, research directed by Dept. of Electrical and Computer Engineering. University of Maryland, College Park).

APPENDIX

- **Source Code**

```
clc

clear all

PATTERNS = [];

PATTERN1 =[];

PATTERN2 =[];

pattern3 =[];

E =[];

N = 100;

IMAGES = cell(1,N);

FNAMEFMT = 'a%d.pgm';


row=100;

colum=100;

% Load images

for i=1:N

IMAGES{i} = imread(sprintf(FNAMEFMT, i));

img=imresize(IMAGES{i},[row,colum]);

img=im2double(img);

origin=img;
```

```
%img=medfilt2(img);

% toc

%% Scale-Space Extrema Detection

tic

% original sigma and the number of actave can be modified. the larger

% sigma0, the more quickly-smooth images

sigma0=sqrt(2);

octave=3;%6*sigma*k^(octave*level)<=min(m,n)/(2^(octave-2))

level=3;

D=cell(1,octave);

for i=1:octave

D(i)=mat2cell(zeros(row*2^(2-i)+2,colum*2^(2-i)+2,level),row*2^(2-i)+2,colum*2^(2-
i)+2,level);

end

% first image in first octave is created by interpolating the original one.

temp_img=kron(img,ones(2));

temp_img=padarray(temp_img,[1,1],'replicate');

figure(2)

subplot(1,2,1);

imshow(origin)

%create the DoG pyramid.
```

```
for i=1:octave
    temp_D=D{i};
    for j=1:level
        scale=sigma0*sqrt(2)^(1/level)^((i-1)*level+j);
        p=(level)*(i-1);
        figure(1);
        subplot(octave,level,p+j);
        f=fspecial('gaussian',[1,floor(6*scale)],scale);
        L1=temp_img;
        if(i==1&&j==1)
        L2=conv2(temp_img,f,'same');
        L2=conv2(L2,f','same');
        temp_D(:,:,j)=L2-L1;
        imshow(uint8(255 * mat2gray(temp_D(:,:,j))));
        L1=L2;
        else
        L2=conv2(temp_img,f,'same');
        L2=conv2(L2,f','same');
        temp_D(:,:,j)=L2-L1;
        L1=L2;
        if(j==level)
```

```
            temp_img=L1(2:end-1,2:end-1);
        end
        imshow(uint8(255 * mat2gray(temp_D(:,:,j))));
        end
    end
    D{i}=temp_D;
    temp_img=temp_img(1:2:end,1:2:end);
    temp_img=padarray(temp_img,[1,1],'both','replicate');
end
toc
%% Keypoint Localistaion
% search each pixel in the DoG map to find the extreme point
tic
interval=level-1;
number=0;
for i=2:octave+1
    number=number+(2^(i-octave)*colum)*(2*row)*interval;
end
extrema=zeros(1,4*number);
flag=1;
for i=1:octave
```

```
[m,n,~]=size(D{i});
m=m-2;
n=n-2;
volume=m*n/(4^(i-1));
for k=2:interval
    for j=1:volume
        % starter=D{i}(x+1,y+1,k);
        x=ceil(j/n);
        y=mod(j-1,m)+1;
        sub=D{i}(x:x+2,y:y+2,k-1:k+1);
        large=max(max(max(sub)));
        little=min(min(min(sub)));
        if(large==D{i}(x+1,y+1,k))
            temp=[i,k,j,1];
            extrema(flag:(flag+3))=temp;
            flag=flag+4;
        end
        if(little==D{i}(x+1,y+1,k))
            temp=[i,k,j,-1];
            extrema(flag:(flag+3))=temp;
            flag=flag+4;
```

```
            end
        end
    end
end
idx= extrema==0;
extrema(idx)=[];
toc
[m,n]=size(img);
x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;
y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;
ry=y./2.^(octave-1-extrema(1:4:end));
rx=x./2.^(octave-1-extrema(1:4:end));
figure(2)
subplot(1,2,2);
imshow(origin)
hold on
plot(ry,rx,'r+');

%% accurate keypoint localization
%eliminate the point with low contrast or poorly localised on an edge
% x:|,y:-- x is for vertial and y is for horizontal
```

```
% value comes from the paper.

tic

threshold=0.1;

r=10;

extr_volume=length(extrema)/4;

[m,n]=size(img);

secondorder_x=conv2([-1,1;-1,1],[-1,1;-1,1]);

secondorder_y=conv2([-1,-1;1,1],[-1,-1;1,1]);

for i=1:octave

    for j=1:level

        test=D{i}(:,:,j);

        temp=-1./conv2(test,secondorder_y,'same').*conv2(test,[-1,-1;1,1],'same');

        D{i}(:,:,j)=temp.*conv2(test',[-1,-1;1,1],'same')*0.5+test;

    end

end

for i=1:extr_volume

    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;

    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;

    rx=x+1;

    ry=y+1;

    rz=extrema(4*(i-1)+2);
```

```
    z=D{extrema(4*(i-1)+1)}(rx,ry,rz);
    if(abs(z)<threshold)
        extrema(4*(i-1)+4)=0;
    end
end
idx=find(extrema==0);
idx=[idx,idx-1,idx-2,idx-3];
extrema(idx)=[];
extr_volume=length(extrema)/4;
x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;
y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;
ry=y./2.^(octave-1-extrema(1:4:end));
rx=x./2.^(octave-1-extrema(1:4:end));
figure(2)
subplot(2,2,3);
imshow(origin)
hold on
w=plot(ry,rx,'g+');
figure('visible','off');
% saveas(w, '1234-visible', 'png' )
```

```
% imwrite(img1,strcat('SegIMG',num2str(k),'.jpg'));

for i=1:extr_volume
   x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;
   y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;
   rx=x+1;
   ry=y+1;
   rz=extrema(4*(i-1)+2);
      Dxx=D{extrema(4*(i-1)+1)}(rx-1,ry,rz)+D{extrema(4*(i-1)+1)}(rx+1,ry,rz)-2*D{extrema(4*(i-1)+1)}(rx,ry,rz);
      Dyy=D{extrema(4*(i-1)+1)}(rx,ry-1,rz)+D{extrema(4*(i-1)+1)}(rx,ry+1,rz)-2*D{extrema(4*(i-1)+1)}(rx,ry,rz);
      Dxy=D{extrema(4*(i-1)+1)}(rx-1,ry-1,rz)+D{extrema(4*(i-1)+1)}(rx+1,ry+1,rz)-D{extrema(4*(i-1)+1)}(rx-1,ry+1,rz)-D{extrema(4*(i-1)+1)}(rx+1,ry-1,rz);
      deter=Dxx*Dyy-Dxy*Dxy;
      R=(Dxx+Dyy)/deter;
      R_threshold=(r+1)^2/r;
      if(deter<0||R>R_threshold)
         extrema(4*(i-1)+4)=0;
      end
```

```
end

idx=find(extrema==0);

idx=[idx,idx-1,idx-2,idx-3];

extrema(idx)=[];

extr_volume=length(extrema)/4;

x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;

y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;

ry=y./2.^(octave-1-extrema(1:4:end));

rx=x./2.^(octave-1-extrema(1:4:end));

figure(2)

subplot(2,2,4);

imshow(origin)

hold on

plot(ry,rx,'b+');

toc

%% Orientation Assignment(Multiple orientations assignment)

tic

kpori=zeros(1,36*extr_volume);

minor=zeros(1,36*extr_volume);

f=1;

flag=1;
```

```
for i=1:extr_volume
    %search in the certain scale
    scale=sigma0*sqrt(2)^(1/level)^((extrema(4*(i-1)+1)-1)*level+(extrema(4*(i-1)+2)));
    width=2*round(3*1.5*scale);
    count=1;
    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;
    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;
    %make sure the point in the searchable area
    if(x>(width/2)&&y>(width/2)&&x<(m/2^(extrema(4*(i-1)+1)-2)-width/2-
2)&&y<(n/2^(extrema(4*(i-1)+1)-2)-width/2-2))
        rx=x+1;
        ry=y+1;
        rz=extrema(4*(i-1)+2);
        reg_volume=width*width;%3? thereom
        % make weight matrix
        weight=fspecial('gaussian',width,1.5*scale);
        %calculate region pixels' magnitude and region orientation
        reg_mag=zeros(1,count);
        reg_theta=zeros(1,count);
    for l=(rx-width/2):(rx+width/2-1)
        for k=(ry-width/2):(ry+width/2-1)
```

```
            reg_mag(count)=sqrt((D{extrema(4*(i-1)+1)}(l+1,k,rz)-D{extrema(4*(i-1)+1)}(l-
1,k,rz))^2+(D{extrema(4*(i-1)+1)}(l,k+1,rz)-D{extrema(4*(i-1)+1)}(l,k-1,rz))^2);
            reg_theta(count)=atan2((D{extrema(4*(i-1)+1)}(l,k+1,rz)-D{extrema(4*(i-1)+1)}(l,k-
1,rz)),(D{extrema(4*(i-1)+1)}(l+1,k,rz)-D{extrema(4*(i-1)+1)}(l-1,k,rz)))*(180/pi);
            count=count+1;
        end
    end
    %make histogram
    mag_counts=zeros(1,36);
    for x=0:10:359
        mag_count=0;
       for j=1:reg_volume
           c1=-180+x;
           c2=-171+x;
           if(c1<0||c2<0)
           if(abs(reg_theta(j))<abs(c1)&&abs(reg_theta(j))>=abs(c2))
               mag_count=mag_count+reg_mag(j)*weight(ceil(j/width),mod(j-1,width)+1);
           end
           else
               if(abs(reg_theta(j)>abs(c1)&&abs(reg_theta(j)<=abs(c2))))
                   mag_count=mag_count+reg_mag(j)*weight(ceil(j/width),mod(j-1,width)+1);
               end
```

```
            end
        end
            mag_counts(x/10+1)=mag_count;
    end
    % find the max histogram bar and the ones higher than 80% max
    [maxvm,~]=max(mag_counts);
     kori=find(mag_counts>=(0.8*maxvm));
     kori=(kori*10+(kori-1)*10)./2-180;
     kpori(f:(f+length(kori)-1))=kori;
     f=f+length(kori);
     temp_extrema=[extrema(4*(i-1)+1),extrema(4*(i-1)+2),extrema(4*(i-1)+3),extrema(4*(i-
1)+4)];
     temp_extrema=padarray(temp_extrema,[0,length(temp_extrema)*(length(kori)-
1)],'post','circular');
     long=length(temp_extrema);
     minor(flag:flag+long-1)=temp_extrema;
     flag=flag+long;
    end
end
idx= minor==0;
minor(idx)=[];
extrema=minor;
```

```
% delete unsearchable points and add minor orientation points
idx= kpori==0;
kpori(idx)=[];
extr_volume=length(extrema)/4;
toc
%% keypoint descriptor
tic
d=4;% In David G. Lowe experiment,divide the area into 4*4.
pixel=4;
feature=zeros(d*d*8,extr_volume);
for i=1:extr_volume
    descriptor=zeros(1,d*d*8);% feature dimension is 128=4*4*8;
    width=d*pixel;
    %x,y centeral point and prepare for location rotation
    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;
    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;
    z=extrema(4*(i-1)+2);
        if((m/2^(extrema(4*(i-1)+1)-2)-
pixel*d*sqrt(2)/2)>x&&x>(pixel*d/2*sqrt(2))&&(n/2^(extrema(4*(i-1)+1)-2)-
pixel*d/2*sqrt(2))>y&&y>(pixel*d/2*sqrt(2)))
        sub_x=(x-d*pixel/2+1):(x+d*pixel/2);
        sub_y=(y-d*pixel/2+1):(y+d*pixel/2);
```

```
    sub=zeros(2,length(sub_x)*length(sub_y));
    j=1;
    for p=1:length(sub_x)
        for q=1:length(sub_y)
            sub(:,j)=[sub_x(p)-x;sub_y(q)-y];
            j=j+1;
        end
    end
    distort=[cos(pi*kpori(i)/180),-
sin(pi*kpori(i)/180);sin(pi*kpori(i)/180),cos(pi*kpori(i)/180)];
  %accordinate after distort
    sub_dis=distort*sub;
    fix_sub=ceil(sub_dis);
    fix_sub=[fix_sub(1,:)+x;fix_sub(2,:)+y];
    patch=zeros(1,width*width);
    for p=1:length(fix_sub)
    patch(p)=D{extrema(4*(i-1)+1)}(fix_sub(1,p),fix_sub(2,p),z);
    end
    temp_D=(reshape(patch,[width,width]))';
    %create weight matrix.
    mag_sub=temp_D;
```

```
temp_D=padarray(temp_D,[1,1],'replicate','both');

weight=fspecial('gaussian',width,width/1.5);

mag_sub=weight.*mag_sub;

theta_sub=atan((temp_D(2:end-1,3:1:end)-temp_D(2:end-1,1:1:end-
2))./(temp_D(3:1:end,2:1:end-1)-temp_D(1:1:end-2,2:1:end-1)))*(180/pi);

% create orientation histogram

for area=1:d*d

cover=pixel*pixel;

ori=zeros(1,cover);

magcounts=zeros(1,8);

for angle=0:45:359

  magcount=0;

  for p=1:cover;

      x=(floor((p-1)/pixel)+1)+pixel*floor((area-1)/d);

      y=mod(p-1,pixel)+1+pixel*(mod(area-1,d));

      c1=-180+angle;

      c2=-180+45+angle;

      if(c1<0||c2<0)

         if (abs(theta_sub(x,y))<abs(c1)&&abs(theta_sub(x,y))>=abs(c2))


            ori(p)=(c1+c2)/2;
```

```
            magcount=magcount+mag_sub(x,y);
        end
      else
        if(abs(theta_sub(x,y))>abs(c1)&&abs(theta_sub(x,y))<=abs(c2))
            ori(p)=(c1+c2)/2;
            magcount=magcount+mag_sub(x,y);
        end
      end
  end
  magcounts(angle/45+1)=magcount;
end
descriptor((area-1)*8+1:area*8)=magcounts;
end
descriptor=normr(descriptor);
% cap 0.2
for j=1:numel(descriptor)
   if(abs(descriptor(j))>0.2)
   descriptor(j)=0.2;
   end
end
descriptor=normr(descriptor);
```

```
    else
       continue;
    end
    feature(:,i)=descriptor';
end
v=descriptor;
vector_photo = reshape(v,[], 1);
PATTERNS =[PATTERNS vector_photo];
end;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CREATING AND INITIATING THE NETWORK
train_input=PATTERNS;
TARGETS=[1 1	1	1	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0
```

0	0	0	0	0	1	1	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									

0	0	0	0	0	0	0	0	0	0	1	1	1
	1	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									

0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									

0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0

0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1
1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	1	1	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	1	1	1
	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									

0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	1	1	1	1	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	1
	1	1	1	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	1	1
	1	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0									
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0

```
        0   0   0   0   0   0   0   0   0   0   0   0
        1   1   1   1   1   0   0   0   0   0   0   0
        0   0   0

0       0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   1   1   1   1   1   0   0
        0   0   0

0       0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   1   1
        1   1   1

];

net = newff(minmax(train_input),TARGETS,[100 20],{'logsig','logsig'},'traingdm');

% TRAINING THE NETWORK

net.trainParam.lr = 0.47;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =6000;% Maximum number of epochs to train.
```

```
net.trainParam.mc = 0.85 % Momentum Factor.

target=[];


[net,tr] = train(net,train_input,TARGETS);


train = sim(net,train_input)

%%%%%%%%%%%%%%%Test the Network


IMAGES = cell(1,N);

FNAMEFMT = 'b%d.pgm';

row=100;

colum=100;

% Load images

for i=1:N

IMAGES{i} = imread(sprintf(FNAMEFMT, i));

img=imresize(IMAGES{i},[row,colum]);

img=im2double(img);

origin=img;

%img=medfilt2(img);

% toc

%% Scale-Space Extrema Detection

tic

% original sigma and the number of actave can be modified. the larger

% sigma0, the more quickly-smooth images
```

```
sigma0=sqrt(2);

octave=3;%6*sigma*k^(octave*level)<=min(m,n)/(2^(octave-2))

level=3;

D=cell(1,octave);

for i=1:octave

D(i)=mat2cell(zeros(row*2^(2-i)+2,colum*2^(2-i)+2,level),row*2^(2-i)+2,colum*2^(2-
i)+2,level);

end

% first image in first octave is created by interpolating the original one.

temp_img=kron(img,ones(2));

temp_img=padarray(temp_img,[1,1],'replicate');

figure(2)

subplot(1,2,1);

imshow(origin)

%create the DoG pyramid.

for i=1:octave

  temp_D=D{i};

  for j=1:level

    scale=sigma0*sqrt(2)^(1/level)^((i-1)*level+j);

    p=(level)*(i-1);

    figure(1);

    subplot(octave,level,p+j);

    f=fspecial('gaussian',[1,floor(6*scale)],scale);

    L1=temp_img;

    if(i==1&&j==1)
```

```
        L2=conv2(temp_img,f,'same');
        L2=conv2(L2,f','same');
        temp_D(:,:,j)=L2-L1;
        imshow(uint8(255 * mat2gray(temp_D(:,:,j))));
        L1=L2;
        else
        L2=conv2(temp_img,f,'same');
        L2=conv2(L2,f','same');
        temp_D(:,:,j)=L2-L1;
        L1=L2;
        if(j==level)
            temp_img=L1(2:end-1,2:end-1);
        end
        imshow(uint8(255 * mat2gray(temp_D(:,:,j))));
        end
    end
    D{i}=temp_D;
    temp_img=temp_img(1:2:end,1:2:end);
    temp_img=padarray(temp_img,[1,1],'both','replicate');
end
toc
%% Keypoint Localistaion
% search each pixel in the DoG map to find the extreme point
tic
interval=level-1;
```

```
number=0;
for i=2:octave+1
    number=number+(2^(i-octave)*colum)*(2*row)*interval;
end
extrema=zeros(1,4*number);
flag=1;
for i=1:octave
    [m,n,~]=size(D{i});
    m=m-2;
    n=n-2;
    volume=m*n/(4^(i-1));
    for k=2:interval
        for j=1:volume
            % starter=D{i}(x+1,y+1,k);
            x=ceil(j/n);
            y=mod(j-1,m)+1;
            sub=D{i}(x:x+2,y:y+2,k-1:k+1);
            large=max(max(max(sub)));
            little=min(min(min(sub)));
            if(large==D{i}(x+1,y+1,k))
                temp=[i,k,j,1];
                extrema(flag:(flag+3))=temp;
                flag=flag+4;
            end
            if(little==D{i}(x+1,y+1,k))
```

```
            temp=[i,k,j,-1];
            extrema(flag:(flag+3))=temp;
            flag=flag+4;
          end
        end
    end
end
idx= extrema==0;
extrema(idx)=[];
toc
[m,n]=size(img);
x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;
y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;
ry=y./2.^(octave-1-extrema(1:4:end));
rx=x./2.^(octave-1-extrema(1:4:end));
figure(2)
subplot(1,2,2);
imshow(origin)
hold on
plot(ry,rx,'r+');

%% accurate keypoint localization
%eliminate the point with low contrast or poorly localised on an edge
% x:|,y:-- x is for vertial and y is for horizontal
% value comes from the paper.
```

```
tic
threshold=0.1;
r=10;
extr_volume=length(extrema)/4;
[m,n]=size(img);
secondorder_x=conv2([-1,1;-1,1],[-1,1;-1,1]);
secondorder_y=conv2([-1,-1;1,1],[-1,-1;1,1]);
for i=1:octave
    for j=1:level
        test=D{i}(:,:,j);
        temp=-1./conv2(test,secondorder_y,'same').*conv2(test,[-1,-1;1,1],'same');
        D{i}(:,:,j)=temp.*conv2(test',[-1,-1;1,1],'same')*0.5+test;
    end
end
for i=1:extr_volume
    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;
    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;
    rx=x+1;
    ry=y+1;
    rz=extrema(4*(i-1)+2);
    z=D{extrema(4*(i-1)+1)}(rx,ry,rz);
    if(abs(z)<threshold)
        extrema(4*(i-1)+4)=0;
    end
end
```

```
idx=find(extrema==0);

idx=[idx,idx-1,idx-2,idx-3];

extrema(idx)=[];

extr_volume=length(extrema)/4;

x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;

y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;

ry=y./2.^(octave-1-extrema(1:4:end));

rx=x./2.^(octave-1-extrema(1:4:end));

figure(2)

subplot(2,2,3);

imshow(origin)

hold on

w=plot(ry,rx,'g+');

figure('visible','off');

% saveas(w, '1234-visible', 'png' )

% imwrite(img1,strcat('SegIMG',num2str(k),'.jpg'));

for i=1:extr_volume

    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;

    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;

    rx=x+1;

    ry=y+1;

    rz=extrema(4*(i-1)+2);

        Dxx=D{extrema(4*(i-1)+1)}(rx-1,ry,rz)+D{extrema(4*(i-1)+1)}(rx+1,ry,rz)-
2*D{extrema(4*(i-1)+1)}(rx,ry,rz);

        Dyy=D{extrema(4*(i-1)+1)}(rx,ry-1,rz)+D{extrema(4*(i-1)+1)}(rx,ry+1,rz)-
2*D{extrema(4*(i-1)+1)}(rx,ry,rz);
```

```
    Dxy=D{extrema(4*(i-1)+1)}(rx-1,ry-1,rz)+D{extrema(4*(i-1)+1)}(rx+1,ry+1,rz)-D{extrema(4*(i-1)+1)}(rx-1,ry+1,rz)-D{extrema(4*(i-1)+1)}(rx+1,ry-1,rz);
    deter=Dxx*Dyy-Dxy*Dxy;
    R=(Dxx+Dyy)/deter;
    R_threshold=(r+1)^2/r;
    if(deter<0||R>R_threshold)
      extrema(4*(i-1)+4)=0;
    end

end
idx=find(extrema==0);
idx=[idx,idx-1,idx-2,idx-3];
extrema(idx)=[];
extr_volume=length(extrema)/4;
x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;
y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;
ry=y./2.^(octave-1-extrema(1:4:end));
rx=x./2.^(octave-1-extrema(1:4:end));
figure(2)
subplot(2,2,4);
imshow(origin)
hold on
plot(ry,rx,'b+');
toc
%% Orientation Assignment(Multiple orientations assignment)
```

```
tic

kpori=zeros(1,36*extr_volume);

minor=zeros(1,36*extr_volume);

f=1;

flag=1;

for i=1:extr_volume

    %search in the certain scale

    scale=sigma0*sqrt(2)^(1/level)^((extrema(4*(i-1)+1)-1)*level+(extrema(4*(i-1)+2)));

    width=2*round(3*1.5*scale);

    count=1;

    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;

    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;

    %make sure the point in the searchable area

    if(x>(width/2)&&y>(width/2)&&x<(m/2^(extrema(4*(i-1)+1)-2)-width/2-
2)&&y<(n/2^(extrema(4*(i-1)+1)-2)-width/2-2))

        rx=x+1;

        ry=y+1;

        rz=extrema(4*(i-1)+2);

        reg_volume=width*width;%3? thereom

        % make weight matrix

        weight=fspecial('gaussian',width,1.5*scale);

        %calculate region pixels' magnitude and region orientation

        reg_mag=zeros(1,count);

        reg_theta=zeros(1,count);

    for l=(rx-width/2):(rx+width/2-1)
```

```
      for k=(ry-width/2):(ry+width/2-1)
         reg_mag(count)=sqrt((D{extrema(4*(i-1)+1)}(l+1,k,rz)-D{extrema(4*(i-1)+1)}(l-1,k,rz))^2+(D{extrema(4*(i-1)+1)}(l,k+1,rz)-D{extrema(4*(i-1)+1)}(l,k-1,rz))^2);
         reg_theta(count)=atan2((D{extrema(4*(i-1)+1)}(l,k+1,rz)-D{extrema(4*(i-1)+1)}(l,k-1,rz)),(D{extrema(4*(i-1)+1)}(l+1,k,rz)-D{extrema(4*(i-1)+1)}(l-1,k,rz)))*(180/pi);
         count=count+1;
      end
   end
   %make histogram
   mag_counts=zeros(1,36);
   for x=0:10:359
      mag_count=0;
     for j=1:reg_volume
        c1=-180+x;
        c2=-171+x;
        if(c1<0||c2<0)
        if(abs(reg_theta(j))<abs(c1)&&abs(reg_theta(j))>=abs(c2))
            mag_count=mag_count+reg_mag(j)*weight(ceil(j/width),mod(j-1,width)+1);
        end
        else
            if(abs(reg_theta(j)>abs(c1)&&abs(reg_theta(j)<=abs(c2))))
                mag_count=mag_count+reg_mag(j)*weight(ceil(j/width),mod(j-1,width)+1);
            end
        end
     end
       mag_counts(x/10+1)=mag_count;
```

```
    end
    % find the max histogram bar and the ones higher than 80% max
    [maxvm,~]=max(mag_counts);
     kori=find(mag_counts>=(0.8*maxvm));
     kori=(kori*10+(kori-1)*10)./2-180;
     kpori(f:(f+length(kori)-1))=kori;
     f=f+length(kori);
     temp_extrema=[extrema(4*(i-1)+1),extrema(4*(i-1)+2),extrema(4*(i-1)+3),extrema(4*(i-
1)+4)];
     temp_extrema=padarray(temp_extrema,[0,length(temp_extrema)*(length(kori)-
1)],'post','circular');
     long=length(temp_extrema);
     minor(flag:flag+long-1)=temp_extrema;
     flag=flag+long;
    end
end
idx= minor==0;
minor(idx)=[];
extrema=minor;
% delete unsearchable points and add minor orientation points
idx= kpori==0;
kpori(idx)=[];
extr_volume=length(extrema)/4;
toc
%% keypoint descriptor
tic
```

```
d=4;% In David G. Lowe experiment,divide the area into 4*4.

pixel=4;

feature=zeros(d*d*8,extr_volume);

for i=1:extr_volume

    descriptor=zeros(1,d*d*8);% feature dimension is 128=4*4*8;

    width=d*pixel;

    %x,y centeral point and prepare for location rotation

    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;

    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;

    z=extrema(4*(i-1)+2);

        if((m/2^(extrema(4*(i-1)+1)-2)-
pixel*d*sqrt(2)/2)>x&&x>(pixel*d/2*sqrt(2))&&(n/2^(extrema(4*(i-1)+1)-2)-
pixel*d/2*sqrt(2))>y&&y>(pixel*d/2*sqrt(2)))

        sub_x=(x-d*pixel/2+1):(x+d*pixel/2);

        sub_y=(y-d*pixel/2+1):(y+d*pixel/2);

        sub=zeros(2,length(sub_x)*length(sub_y));

        j=1;

        for p=1:length(sub_x)

            for q=1:length(sub_y)

                sub(:,j)=[sub_x(p)-x;sub_y(q)-y];

                j=j+1;

            end

        end

        distort=[cos(pi*kpori(i)/180),-
sin(pi*kpori(i)/180);sin(pi*kpori(i)/180),cos(pi*kpori(i)/180)];

    %accordinate after distort
```

```
sub_dis=distort*sub;
fix_sub=ceil(sub_dis);
fix_sub=[fix_sub(1,:)+x;fix_sub(2,:)+y];
patch=zeros(1,width*width);
for p=1:length(fix_sub)
patch(p)=D{extrema(4*(i-1)+1)}(fix_sub(1,p),fix_sub(2,p),z);
end
temp_D=(reshape(patch,[width,width]))';
%create weight matrix.
mag_sub=temp_D;
temp_D=padarray(temp_D,[1,1],'replicate','both');
weight=fspecial('gaussian',width,width/1.5);
mag_sub=weight.*mag_sub;
theta_sub=atan((temp_D(2:end-1,3:1:end)-temp_D(2:end-1,1:1:end-
2))./(temp_D(3:1:end,2:1:end-1)-temp_D(1:1:end-2,2:1:end-1)))*(180/pi);
% create orientation histogram
for area=1:d*d
cover=pixel*pixel;
ori=zeros(1,cover);
magcounts=zeros(1,8);
for angle=0:45:359
  magcount=0;
  for p=1:cover;
      x=(floor((p-1)/pixel)+1)+pixel*floor((area-1)/d);
      y=mod(p-1,pixel)+1+pixel*(mod(area-1,d));
```

```
      c1=-180+angle;
      c2=-180+45+angle;
      if(c1<0||c2<0)
          if (abs(theta_sub(x,y))<abs(c1)&&abs(theta_sub(x,y))>=abs(c2))

              ori(p)=(c1+c2)/2;
              magcount=magcount+mag_sub(x,y);
          end
      else
          if(abs(theta_sub(x,y))>abs(c1)&&abs(theta_sub(x,y))<=abs(c2))
              ori(p)=(c1+c2)/2;
              magcount=magcount+mag_sub(x,y);
          end
      end
  end
  magcounts(angle/45+1)=magcount;
end
descriptor((area-1)*8+1:area*8)=magcounts;
end
descriptor=normr(descriptor);
% cap 0.2
for j=1:numel(descriptor)
    if(abs(descriptor(j))>0.2)
    descriptor(j)=0.2;
    end
```

```
        end
        descriptor=normr(descriptor);
        else
           continue;
        end
        feature(:,i)=descriptor';
end
v=descriptor;
vector_photo = reshape(v,[], 1);
PATTERNS =[PATTERNS vector_photo];
end;
test_input=PATTERNS;
test = sim(net,test_input)
```