A Multi-Layer Perceptron Neural Network Based-Model for Face Detection

By:

(Ghaith Ahmad Al-Qudah)

Supervised By:
(Prof. Nidal Shilbayeh)

Master Thesis

Submitted in Fulfillment of the Requirements for the Degree of Master of Science In Computer Science

Middle East University for Graduate Studies (MEU)

May, 2009

نموذج تفويض

أنا الطالب (**غيث احمد القضاة**) أفوض جامعة الشرق الأوسط للدراسات العليا بتزويد نسخ من رسالتي للمكتبات أو الهيئات أو الأفراد عند طلبها .

التوقيع :-
التاريخ :- 2009 / 6 / 1

Middle East University for Graduate Studies

Authorized Form

I (Ghaith Ahmad Al-Qudah), authorize the Middle East University for Graduate Studies to supply copies of my thesis to libraries, establishments or individuals on request.

Signature:-
Date: - 1 / 6 / 2009

Middle East University

Examination Committee Decision

"This is to certify that the Thesis entitled" A Multi-Layer Perceptron Neural Network Based-Model for Face Detection was successfully defended and approved on May 31st 2009"

Examination Committee Members	**Signature**
Prof. Nidal shilbaeh Prof. Department of Computer Information System . (Middle East University)	…………………....
Prof. Rustom Al- Mamlook Prof. Department of Computer Information System (Middle East University)	Rustom Mamlook
Dr. Hussein Hadi Owaied Associate Prof. Department of Computer Science (Middle East University)	…………………… Owaied
Prof. Waleed salameh . Prof. Department of Computer Networks and Information System (New York Institute of Technology)	Waleed Salameh

ACKNOWLEDGEMENT

Firstly, I would like to express my gratitude to my advisor Prof. Dr. Nidal Shilbayeh for his good guidance, support and attention from the beginning till the end. Also thanks and extended to my parent for their support and motivation in master education and thesis, even before the beginning. I thank friends in Image Processing for sharing their experience and knowledge.

Abstract

We propose a face detector using an efficient architecture based on a multi layer-perceptron neural network (MLPNN) with maximal rejection classifier (MRC) which we achieve enhanced detection accuracy and efficiency using matlab. First, experimental results show that the proposed neural-network significantly improves the detection accuracy as compared to traditional neural-network-based techniques. Second, in order to reduce the total computation cost for the face detection, we organize the neural network in this way, simply used at earlier stages are able to reject a majority of non-face patterns in the image backgrounds, thereby significantly improving the overall detection efficiency while maintaining the detection accuracy. An important advantage of the new architecture is that it has a homogeneous structure so that it is suitable for very efficient implementation using programmable devices. Our proposed approach achieves one of the best detection accuracies with significantly reduced training and detection cost. The goal of this project is to detect and locate human faces in a color image. A set of training images were provided for this purpose. The objective was to design and implement a face detector in MATLAB and Visual Studio .NET that will detect human faces in an image similar to the training images.

Key words: **Face Detection, Multi-Layer Perceptron (MLP), Learning, Neural Networks, Maximal Rejection Classifier (MRC).**

خلاصة

نقوم بتحديد الوجه باستخدام فعال يقوم على بنية متعددة الطبقات في الشبكة العصبية (MLPNN) إلى أقصى حد مع استخدام طريقة الرفض الأكثر (Maximal Rejection Classifier MRC) الذي يعزز دقة وكفاءة استخدام النظام.

أولا، تظهر النتائج التجريبية في الشبكة العصبية تحسن كبير في الكشف عن الوجه من حيث الدقة مقارنة بالشبكة العصبية التقليدية القائمة بدون استخدام إل Maximal rejection Classifier. ثانيا ، من أجل تقليل التكاليف الإجمالية الحسابية للكشف عن وجهه ، ونحن في تنظيم الشبكة العصبية في هذا السبيل ، وببساطة فمنا باستخدام المبسط في المراحل الأولى من النظام، حيث تم إزالة أغلبية الخلفيات والمواقع التي لا تظهر صورة الوجه في الصورة ، مما يؤدي إلى تحسين كبير في الكفاءة والمحافظة في الوقت نفسه على الدقة.

ميزة هامة في النظام الجديد هو أن هذا النظام متجانس بحيث يكون ملائما للتنفيذ الفعال لغاية استخدام الكشف عن الأشياء الأخرى مثل الكشف عن السيارات المطلوبة والمباني الخ. نظامنا يحقق واحدا من أفضل الأنظمة الموجودة حاليا مع دقة في النتائج وانخفاض كبير في تكاليف الاكتشاف.

خلاصة هذا المشروع هو اكتشاف وتحديد الأوجه البشرية في الصور الملونة بغض النظر عن خلفية الصورة من ناحية التعقيد فيها. قمنا بتصميم وتنفيذ البرنامج باستخدام أل MATLAB وال Microsoft.NET وحصلنا على نتيجة 92% من الهدف المطلوب.

الكلمات الأساسية:

Face Detection, Multi-Layer Perceptron (MLP), Learning, Neural Networks, Maximal Rejection Classifier (MRC).

Table of Contents (TOC)

LIST OF TABLES

LIST OF Figures

List of Abbreviations

MLP	Multi layer perceptron.
MRC	Maximal Rejection Classifier
ANN	Artificial Neural Network
HMM	Hidden Markov Model
SVM	Support Vector Machines

CHAPTER 1

INTRODUCTION

Nowadays, there is an increasing trend of using biometrics information, which refers the human biological features used for user authentication, such as fingerprint, iris, and face. Face detection has attracted many researchers because it has a wide area of applications. Given an image, face detection involves localizing all faces - if any - in this image. Face detection is the first step in any automated system that solves problems such as: face recognition, face tracking and facial expression recognition.

Face detection is the first step before face recognition. Its reliability and performance have a major influence in a whole pattern recognition system. Nowadays, neural networks have shown very good results for detecting a certain pattern in a given image [1, 2, 3]. But the problem with neural networks is that the computational complexity is very high because the networks have to process many small local windows in the images [4, 5]. Some authors tried to speed up the detection process of neural networks [6, 7, 8]. They proposed a multilayer perceptron (MLP) algorithm for fast object/face detection.

They claimed that applying cross-correlation in the frequency domain between the input image and the neural weights is much faster than using conventional neural networks. They stated this without any conditions and introduced formulas for the number of computation steps needed by conventional neural networks and their proposed fast neural networks.

Face detection is becoming one of the most active research areas in computer vision and a very important research topic in biometrics; due to its wide range of possible applications, like security access control, video surveillance and image database management, and it is the first step to face recognition System.

The face detection is to identify the presence of a human face in an image. Face Detection, looks at images, and tries to make a judgment on whether or not that image contains a face.

1.1 Problem definition

Face detection is the problem of determining whether there are human faces in the image, and if there are, returning the location of each human face in the image. Face detection is the most important step in any face recognition system. So far, the researchers have mainly focused on the face recognition, in which the task of finding faces in an arbitrary background is a usually avoided by segmentation of the input images or by capturing faces against a known uniform background. Face detection also has potential applications in human computer interface and surveillance systems.

Variations in images increase the complexity of the decision boundary between face and non face classes. Such complexity will also increase the difficulty of the problem and make the classification to be harder. These variations can be summarized as follows [4 ,9 ,12]:

- Pose: the image of face varies due to the relative face pose (frontal, 45 degree, profile, upside down), and some facial features such as eye or the nose may become partially or wholly occluded. Example of such variations is shown in Figure 1.1.

- Illumination: same face, with the same facial expression, and seen from the same viewpoint appears differently due to changes in lighting. Example of such variations is shown in Figure 1.2
- Facial Expressions: the appearance of faces is directly affected by a person's facial expression. Example of such variations is shown in Figure 1.3.
- Presence or absence of structural components: facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, color, and size. Example of such variations is shown in Figure 1.4.
- Occlusion: faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces. Example of such variations is shown in Figure 1.5.
- Image Orientation: face images directly vary for different rotations about the axis. Example of such variations is shown in Figure 1.6.
- **Image Conditions:** when the image is formed, variations such as lighting and camera characteristics affect the appearance of a face. Example of such variations is shown in Figure 1.7.

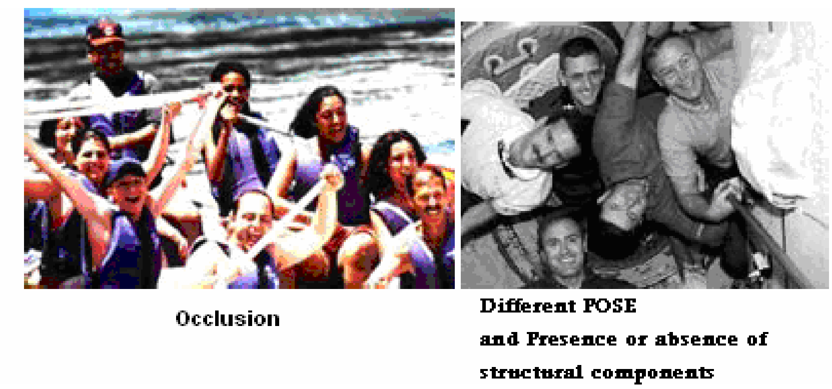


Figure 1.1: Different Image Conditions

Figure 1.2: Different Image Illumination

Figure 1.3: Different Facial Expression

1.2 Thesis objectives

Reviewing the literature, it is quite difficult to state that there exists a complete system which solves face detection problem with all variations included. The main objective of this thesis is to design and implement face detection system that locate upright frontal faces on digital color images with high detection rate. Also, the system able to detect faces of various sizes in complex background images. We proposed a new method for improving the detection rate through the use of classifier cascade. The elegant key behind the cascade is based on a tree structure of classifiers. In detecting task in a large image a large majority of the sub-windows observed by scanning the image using the MRC classifier will be rejected and just a small regional area(s) in the image might be targeted. Therefore, the generality of the first stage must be sufficiently high to reduce the number of these false positive sub-windows from processing in the next stage using MLP neural network. The aim of cascading is to provide inference with the lowest possible false positives, and highest detection rate. In this study, for decreasing variations in the training examples, pose variations are limited and profile and excessive rotated faces are excluded.

1.3 Thesis Organization

The remainder of this thesis is organized as follows:

In Chapter 2, the overview of face detection methods in the literature is given.

Chapter 3, provides a description of the proposed method used in this thesis. The new method uses a cascade classifier that includes Maximal rejection Method (MRC) and MLP Neural Networks.

Chapter 4, describes the simulation results from our utilized two classifiers and comparing these results with the previous work.

Chapter 5, summarizes the conclusions of this study. Recommendations for future works are provided in this chapter.

Chapter 2
Literature survey

Over the last twenty years, there has been a great deal of research concerning important aspects of face detection. Face detection research can be heuristically classified in two main categories **Feature-based approaches & Image-based approaches** as shown in Figure 2.1.

All face detection techniques require a priori knowledge of the face. Each category will be explained, and the work completed will be presented, providing a brief of the various face detection techniques.

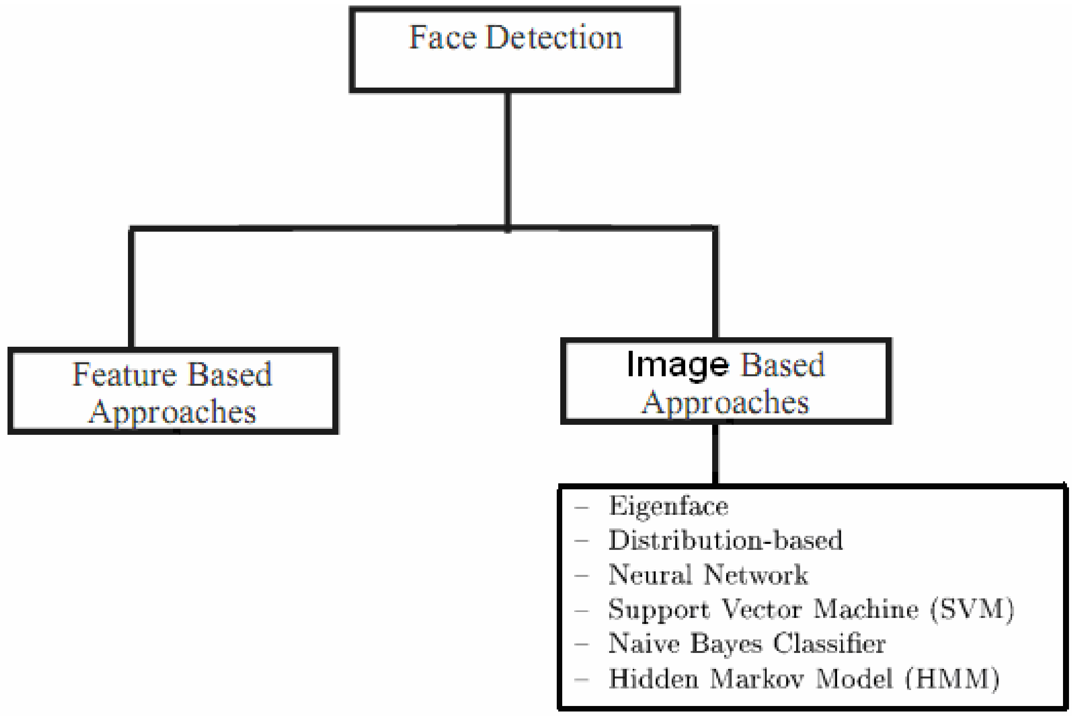


Figure 2: Face Detection Methods

2.1 Feature-Based Approach

It depends on feature derivation and analysis to gain the required knowledge about faces. It consists of three sub-categories: low-level analysis, Template Matching analysis, and generalized Knowledge Rules. Features may be skin color, face shape, or facial features like eyes, nose, etc. This area contains techniques that are classified **as low-level analysis**.

These methods deal with the segmentation of visual features using pixel properties such as gray-scale and color. The features that these low-level methods detect can be ambiguous, but these methods are easy to implement and fast.

Feature based methods are preferred for real time systems where the multi-resolution window scanning used by image based methods are not applicable. On the other hand, image-based techniques treat face detection as a general pattern recognition problem. It uses training algorithms to classify regions into face or non-face classes. A variety of methods that use low-level analysis in face detection, such as edges, color segmentation are done by, [35,37].

Feature analysis face detection is based upon facial features using information of face geometry. Through feature analysis, feature ambiguities are reduced and location of the face and facial features are determined. The feature analysis method uses a group of active shape models. This method depends on feature searching and works on the premise that is exclusively looks for prominent facial features in the image. This method is restricted to frontal face images with plain background and it needs a clear forehead not hidden by hair to insure detection. If facial hair, earning and eyewear are worn on the faces it fails to detect the face [37].

2.2 Image-Based Approach

Most image-based approaches apply a window scanning technique for detecting faces, which due to it's exhaustive nature, increases computational demand. It depends on multi-resolution window scanning to detect faces, so these techniques have high detection rates but slower than the feature-based techniques. The image-based approach is usually limited to detecting one face in a non-complex background with ideal conditions. There is a need for techniques that can detect multiple faces. This technique works on the idea that the face is recognized by comparing an image with examples of face patterns. This eliminates the use of face knowledge as the detection technique. This means inaccurate or uncompleted data from facial images can still be detected as a face.

The approach here is to classify an area as either face or non-face, so a set of face and non face prototypes must be trained to fit these patterns. We present the techniques that use a feature based approach. They are *Eigenfaces, Distribution Based, Neural Networks, Support Vector Machines*, Hidden Markov Model And information theoretical approach [6, 21].

2.2.1 Eigenfaces: A technique using principal component analysis to efficiently represent human faces. Given an ensemble of different face images, the technique first finds the principal components of the distribution of faces, expressed in terms of eigenvectors (taken from a 2D image matrix). Each individual face in the face set can then be approximated by a linear combination of the largest eigenvectors, more commonly referred to as eigenfaces, using appropriate weights. The eigenfaces are determined by performing a principal component analysis on a set of example images with central faces of the same size. In addition the existence of a face in a given image can be determined. By moving a window covering a sub image over the entire image faces can be located within the entire image. A pre compiled set of photographs comprises the training set, and it is this training set that the eigenfaces are extracted from. The photographs in the training set are mapped to another set, which are the eigenfaces. As with any other mapping in mathematics, we can now think of the data (the photographs and the eigenfaces) as existing in two domains. The photographs in the training set are one of these domains, and the eigenfaces comprise the second domain that is often referred to as the face space.

2.2.2 Distribution Based Methods: Sung and Poggio developed a distribution-based system for face detection [15], [35] which demonstrated how the distributions of image patterns from one object class can be learned from positive and negative examples (i.e., images) of that class. Their system consists of two components, distribution-based models for face/nonface patterns and a multilayer perceptron classifier. Each face and nonface example is first normalized and processed to a 19 * 19 pixel image and treated as a 361-dimensional vector or pattern. Next, the patterns are grouped into six face and six nonface clusters using a modified k-means algorithm. Each cluster is represented as a multidimensional Gaussian function with a mean image and a covariance matrix. Two distance metrics are computed between an input image pattern and the prototype clusters. The first distance component is the normalized Mahalanobis distance between the test pattern and the cluster centroid, measured within a lower-dimensional subspace spanned by the cluster's 75 largest eigenvectors. The second distance component is the Euclidean distance between the test pattern and its projection onto the 75-dimensional subspace. This distance component accounts for pattern differences not captured by the first distance component. The last step is to use a multilayer perceptron (MLP) network to classify face window patterns from nonface patterns using the twelve pairs of distances to each face and nonface cluster. The classifier is trained using standard backpropagation formula database of 47,316window patterns. There are 4,150 positive examples of face patterns and the rest are nonface patterns.Note that it is easy to collect a representative sample face patterns, but much more difficult to get a representative sample of nonface patterns. This problem is alleviated by a bootstrap method that selectively adds images to the training set as training progress. Starting with a small set of nonface examples in the training set, the MLP classifier is trained with this database of examples. Then, they run the face detector on a sequence of random images and collect all the nonface patterns that the current system wrongly classifies as faces. These false positives are then added to the training database as new nonface examples. This bootstrap method avoids the problem of explicitly collecting a representative sample of nonface patterns and has been used in later works [41].

2.2.3 Hidden Markov Model: The underlying assumption of the Hidden Markov Model (HMM) is that patterns can be characterized as a parametric random process, and that the parameters of this process can be estimated in a precise well-defined manner. In developing an HMM for a pattern recognition problem, a number of hidden states need to be decided first to form a model. Then, one can train HMM to learn the transitional probability between states from the examples where each example is represented as a sequence of observations. The goal of training an HMM is to maximize the probability of observing the training data by adjusting the parameters in an HMM model. After the HMM has been trained, the output probability of an observation determines the class to which it belongs.

Intuitively, a face pattern can be divided into several regions such as the forehead, eyes, nose, mouth, and chin. A face pattern can then be recognized by a process in which these regions are observed in an appropriate order (from top to bottom and from left to right). Instead of relying on accurate alignment as in template matching or appearance-based methods (where facial features such as eyes and noses need to be aligned well with respect to a reference point), this approach aims to associate facial regions with the states of a continuous density Hidden Markov Model.

HMM-based methods usually treat a face pattern as a sequence of observation vectors where each vector is a strip of pixels. During training and testing, an image is scanned in some order (usually from top to bottom) and an observation is taken as a block of pixels. For face patterns, the boundaries between strips of pixels are represented by probabilistic transitions between states, and the image data within a region is modeled by a multi-variation Gaussian distribution. An observation sequence consists of all intensity values from each block. The output states correspond to the classes to which the observations belong. After the HMM has been trained, the output probability of an observation determines the class to which it belongs. HMM have been applied to both face recognition and localization**.**

2.2.4 Support Vector Machines (SVM): Support vector machine is a patter classification algorithm developed by V. Vapnik and his team at AT&T Bell Labs [6]. While most machine learning based classification techniques are based on the idea of minimizing the error in training data (**empirical risk**) SVM's operate on another induction principle , called ***structural risk minimization***, which minimizes an upper bound on the generalization error. Training is performed with a boot-strap learning algorithm [21]. Generating a training set for the SVM is a challenging task because of the difficulty in placing "characteristic" non-face images in the training set. To get a representative sample of face images is not much of a problem; however, to choose the right combination of non-face images from the immensely large set of such images is a complicated task. For this purpose, after each training session, non-faces incorrectly detected as faces are placed in the training set for the next session. This "**bootstrap**" method overcomes the problem of using a huge set of non-face images in the training set, many of which may not influence the training [41]. To test the image for faces, possible face regions detected by another technique (say, color segmentation) will only be tested to avoid exhaustive scanning. In order to explain SVM process consider data points of the form {(xi, yi)} i=1..N. We wish to determine among the infinite such points in an N-dimensional space which of two classes of such points a given point belongs to. If the two classes are linearly separable, we need to determine a hyper-plane that separates these two classes in space. However, if the classes are not clearly separable, then our objective would be to minimize the smallest generalization error. Intuitively, a good choice is the hyper-plane that leaves the maximum margin between the two classes (margin being defined as the sum of the distances of the hyper-plane from the closest points of the two classes), and minimizes the misclassification errors. The same data used to train a neural network can be trained here. The learning time for SVM algorithms are significantly smaller than that for the neural network. Back propagation of a neural network takes more time than the required training time segmentation program to be tested of a SVM training period.

2.2.5 Neural Networks (NN):

Neural networks have been applied successfully in many pattern recognition problems, such as optical character recognition and object recognition. Since face detection can be treated as a two class pattern recognition problem, various neural network architectures have been proposed. The advantage of using neural networks for face detection is the feasibility of training a system to capture the face patterns.

However, one drawback is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance. An early method using hierarchical neural networks was proposed by Rowley et al [41] *(detailed in section 2.2.1)*. Their system operates in two stages: applies first as a set of neural network-based filters to an image and then uses an arbitrator to combine the outputs. The filters examine each location in the image at several scales, looking for locations that might contain a face. The arbitrator then merges detections from individual filters and eliminates overlapping detections. The first component of their system is a filter that receives as input a 20 * 20 pixel region of the image and generates an output ranging from 1 to -1, signifying the presence or absence of a face, respectively. To detect faces anywhere in the input, the filter is applied at every location in the image. To detect faces larger than the window size, the input image is repeatedly reduced in size (by sub sampling), and the filter is applied at each size. Then, histogram equalization is performed; the preprocessed window is then passed through a neural network. The network has retinal connections to its input layer; there are three types of hidden units: four which look at 10 * 10 pixel sub regions, 16 which look at 5 * 5 pixel sub regions and six which look at overlapping 20 * 5 pixels horizontal stripes of pixels. Each of these types was chosen to allow the hidden units to detect local features that might be important for face detection. In particular, the horizontal stripes allow the hidden units to detect such features as mouths or pairs of eyes, while the hidden units with square receptive fields might detect features such as individual eyes, the nose, or corners of the mouth. Finally the detection rate is about 77.9 percent. The limitation of their system is that the images must be frontal faces (Their method cannot detect partially occluded face and the pose).

A different neural approach was suggested by Feraud et al [13] based on a Constrained Generative Model (CGM), an auto associative fully connected MLP with three layers. Lin et al [46], propose a fully automatic face recognition system based on probabilistic decision-based neural networks (PDBNN). For face detection the network has only one subnet representing the face class, and training

is unsupervised. The limitation of his system is that the images must be frontal faces (cannot detect partially occluded face).
Propp and Samal developed one of the earliest neural networks for face detection [40]. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units. A similar hierarchical neural networks is later proposed by [6]. The early method by Soulie et al. [46] scans an input image with a time-delay neural network [14] (with a receptive field of 20 _ 25 pixels) to detect faces. To cope with size variation, the input image is decomposed using wavelet transforms. They reported a false negative rate of 2.7 percent and false positive rate of 0.5 percent from a test of 120 images. In, Vaillant et al. used convolutional neural networks to detect faces in images. Examples of face and nonface images of 20 _ 20 pixels are first created. One neural network is trained to find approximate locations of faces at some scale. Another network is trained to determine the exact position of faces at some scale. Given an image, areas which may contain faces are selected as face candidates by the first network.

These candidates are verified by the second network. Burel and Carel [10] proposed a neural network for face detection in which the large number of training examples of faces and nonfaces are compressed into fewer examples using a Kohonen's SOM algorithm [14]. A multilayer perceptron is used to learn these examples for face/background classification. The detection phase consists of scanning each image at various resolutions. For each location and size of the scanning window, the contents are normalized to a standard size, and the intensity mean and variance are scaled to reduce the effects of lighting conditions. Each normalized window is then classified by an MLP.

Feraud and Bernier presented a detection method using autoassociative neural networks [15], [13], [14]. The idea is based on [30] which show an autoassociative network with five layers is able to perform a nonlinear principal component analysis. One autoassociative network is used to detect frontal-view faces and another one is used to detect faces turned up to 60 degrees to the left and right of the frontal view. A gating network is also utilized to assign weights to frontal and turned face detectors in an ensemble of autoassociative networks. On a small test set of 42 images, they report a detection rate similar to . The method has also been employed in LISTEN [23] and MULTRAK [9]. Lin et al. presented a face detection system using probabilistic decision-based neural network (PDBNN). The architecture of PDBNN is similar to a radial basis function (RBF) network with modified learning rules and probabilistic interpretation. Instead of converting a whole face image into a training vector of intensity values for the neural network, they first extract feature vectors based on intensity and edge information in the facial region that contains eyebrows, eyes, and nose. The extracted two feature vectors are fed into two PDBNN's and the fusion of the outputs determines the classification result. Based on a set of 23 images provided by Sung and Poggio [47], their experimental results show comparable performance with the other leading neural network-based face detectors [11].

Among all the face detection methods that used neural networks, the most significant work is arguably done by Rowley et al. A multilayer neural network is used to learn the face and nonface patterns from face/ nonface images (i.e., the intensities and spatial relationships of pixels) whereas Sung used a neural network to find a discriminant function to classify face and nonface patterns using distance measures. They also used multiple neural networks and several arbitration methods to improve performance, while Burel and Carel [11] used a single network, and Vaillant et al. [5] used two networks for classification. There are two major components: multiple neural networks (to detect face patterns) and a decision making module (to render the final decision from multiple detection results). As shown in Fig. 10, the first component of this method is a neural network that receives a 20 _ 20 pixel region of an image and outputs a score ranging from -1 to 1. Given a test pattern, the output of the trained neural network indicates the evidence for a nonface (close to -1) or face pattern (close to 1). To detect faces anywhere in an image, the neural network is applied at all image locations. To detect faces larger than 20 * 20 pixels, the input image is repeatedly sub sampled, and the network is applied at each scale. Nearly 1,050 face samples of various sizes, orientations, positions, and intensities are used to train the network. In each training image, the eyes, tip of the nose, corners, and center of the mouth are labeled manually and used to normalize the face to the same scale, orientation, and position. The second component of this method is to merge overlapping detection and arbitrate between the outputs of multiple networks. Simple arbitration schemes such as logic operators (AND/OR) and voting are used to improve performance. Rowley et al. [41] reported several systems with different arbitration schemes that are less computationally expensive than Sung and Poggio's system and have higher detection rates based on a test set of 24 images containing 144 faces.

Juell and Marsh proposed a hierarchical neural network to detect human faces in gray scale images. An edge enhancing preprocessor and four back propagation neural networks arranged in a hierarchical structure were devised to find multiple faces in the scene. Their approach was invariant with respect to translation, rotation, and scale, but they cannot classify the pose. Han et al. proposed a system using a morphology-based technique to perform eye-analogue segmentation. Then, a trained back propagation neural network performs the face verification task. Nevertheless, they cannot deal with partially occluded faces.

2.2.6 Summary

In this chapter, we presented a literature review of the general methods currently used to detect faces in images. **Feature-based methods** can be invariant to pose and orientation change. However, it may be difficult to locate all facial features due to the inefficiency of these methods to cope with external factors such as illumination, noise or occlusion. It is also difficult to detect features in complex backgrounds. **Image-based methods** use powerful machine learning algorithms. They have demonstrated accurate, fast and fairly robust empirical results, and can be extended to detect faces in different pose orientations. On the other hand, they usually need to search over space and scale, and require many positive and negative examples.

This research presented in this thesis introduces an image-based face detection method, which can achieve high detection rates, under wider facial conditions.

Chapter 3
Face Detection

3.1. Introduction

Our System is designed to locate multiple faces of 20 * 20 pixel minimum sizes; a practical system is implemented using MATLAB 7.0 and visual studio .NET. Our face detection system consists of several techniques cascaded together to achieve maximal non-face rejection and then detect the faces. The goal to choose the cascading technique is that if we want to detect a specific object in an image, we must check all the pixels of the image which it is an expensive task. This approach is used by all the object detection algorithms. Therefore, a method to increase the detection speed is based on a cascaded evaluation of filters. This is achieved by building a cascade of classifier of increasing complexity by maximal rejection classifier with a multi-layer perceptron neural network (MRC & MLP). In the case of face detection, if a pixel is classified as a non-face at any stage of the cascade, then the pixel is rejected and no further processing is spent on that pixel. Another goal of the cascading presented here is to build a double filtering face detection system for colored images to get more accurate result.

The reason to choose MRC cascaded with MLP Neural Network is that the MRC uses simple (and therefore **fast**) classifiers, another advantage of MRC is simple to implement, and it is also very **accurate** [31, 33, 45]. The neural networks used in this paper multi-layer perceptron (MLP) neural networks [26]. A neural network is created and trained using (MLP) Neural Network, the goal of chosen that type of NN is its **simplicity** and its capability in supervised pattern matching[7,10].

3.2. Description of the System

The face detection problem can be specified as the need to detect all instances of faces in a given image, at all positions, all scales, all facial expressions, of all people, and under all lighting conditions. All these requirements should be met, while having few or no false alarms and miss-detections, and with fast algorithm as possible. In this paper we are using the MRC cascaded with MLP NN for this task because it is **Fast & Accurate Classifier**.

In the first component of our system is a filter that receives as input a 20 * 20 pixel region of the image and generates an output ranging from 1 to -1, signifying the presence or absence of a face, respectively. To detect faces anywhere in the input, the filter is applied at every location in the image. Before we pass the image to the Neural Network we first apply a preprocessing step, adapted from [21], to a window of the image. The window is then passed to another network (MRC) to reject the non face part then pass through a neural network, which decides whether the window contains a face as shown in figure 3.1.

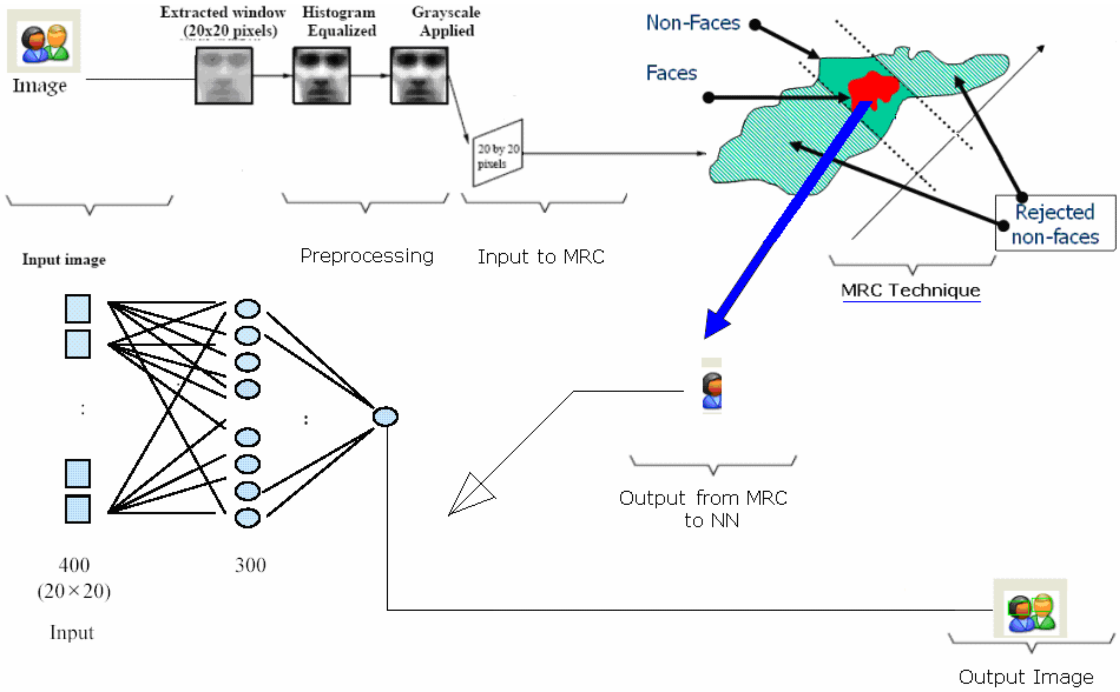


Figure 3.1: Cascading MRC and NN

3.2.1 Cascading MRC and NN

Before feeding the input image to the system as we see in figure 3.2, our face detection system does the following stages:

- Image enhancement & Preprocessing.
- Facial features extraction

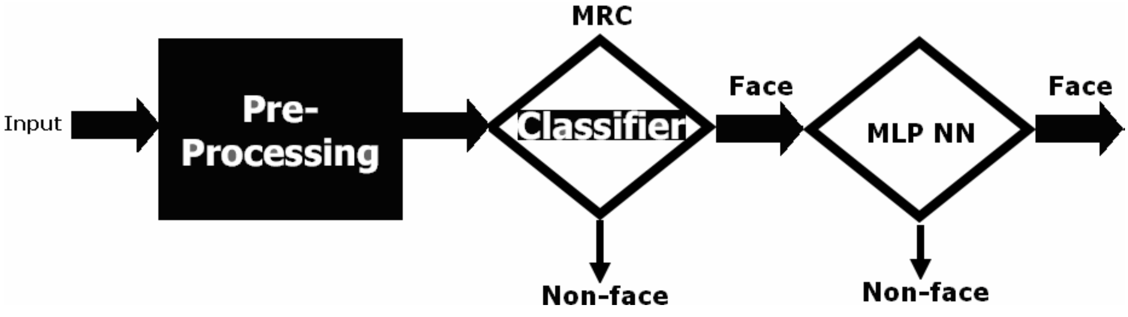


Figure3.2: Cascading MRC and NN Diagram

In an Image enhancement stage, an input image is passed to the system for classification. Image varies in format, size, and resolution. The histogram equalization is performed to enhance the contrast in the image. It compensates changes in illumination in different images. Histogram equalization equitably spreads the pixels of an image among the grey-level intensities thereby increasing the contrast of the image. After that **in preprocessing stage,** the two classes both "non-faces" and "faces" were first subjected to the process of resizing by 20x20 as example and the size will considered to give us the best detection in the following stages. So each image in the "non-face" and "face" class was first resize to 20x20 sizes. Then with the use of standard histogram equalization algorithm each image of both "non-face" and "face" classes were histogram equalized in order to correct brightness, contrast and equalize the different intensities level of the image. After the histogram equalization, the technique of grayscale is applied to the images of both the classes in order to convert their color levels (i.e. RGB) to gray level.

Facial feature extraction stage, 20x20 sized small images are extracted from the reduced sized image and fed to the maximal rejection classifier (MRC) and then the target we do some operations using the MRC and pass the target section to the neural network. In face detection, majority of the proposed methods use image gray level values to detect faces in spite of the fact that most images today are colored to solve these problems, our approach will deal with face detection methods in a color image.

After we do the preprocessing stage on the image we pass it to the MRC network, the classifier (MRC) is used to explore face candidate locations and try to filter as many non-face patterns as possible before passing hard patterns to the final stage classifier; it means false rejection must be avoided at any cost. Because of the property of MRC, we can avoid false rejection as long as we give MRC the enough training data in learning stage. In other words, the training set must be ideally and identically distributed in the sample space. The last stage classifier is a neural network classifier.

The neural network used in our work is the multilayer perceptron (MLP) which is a feed-forward neural network that has been used extensively in classification and regression. The MLP is capable of producing more complex decision boundaries. We use a neural network with the MLP architecture and feed forward topology to classify face and non-face. The employed multilayer feed forward neural net-work consists of neurons with a sigmoid activation function. The employed neural networks are used in two modes. In classification mode, it is presented at the input layer and is propagated forward through the network to compute the activation value for each output neuron. The second mode is called the training or learning mode. Learning in ANN involves the adjustment of the weights in order to achieve the desired processing for a set of learning face samples. More specifically, the second mode includes feeding a neural network with a number of training pairs. Then the networks parameters are adjusted through a supervised training algorithm.

3.3. MRC Face Detection

The Maximal Rejection Classifier (MRC) is a linear classifier that overcomes the two drawbacks. While maintaining the simplicity of a linear classifier, it can also deal with non linearly separable cases. The only requirement is that the Clutter class and the Target class are disjoint. MRC is an iterative rejection based classification algorithm. The main idea is to apply a linear projection followed by a thresholding, at each iteration, just like in the SVM. However, as opposed to these two methods, the projection vector and the corresponding thresholds are chosen such that at each iteration .MRC attempts to maximize the number of rejected Clutter samples. This means that after the first classification iteration, many of the Clutter samples are already classified as such, and discarded from further consideration. The process is continued with the remaining Clutter samples, again searching for a linear projection vector and thresholds that maximizes the rejection of Clutter points from the remaining set. This process is repeated iteratively until convergence to zero or a small number of Clutter points. The remaining samples at the final stage are considered as targets.

Now, that we have segmented every image into several segments and approximated every segment with a small number of representative pixels, we can exhaustively search for the best combination of segments that will reject the largest number of non-face images. We repeat this process until the improvement in rejection is negligible. It means that we must find theta and two decision levels that the number of rejected non-faces is maximized while finding faces. As shown in the figure below

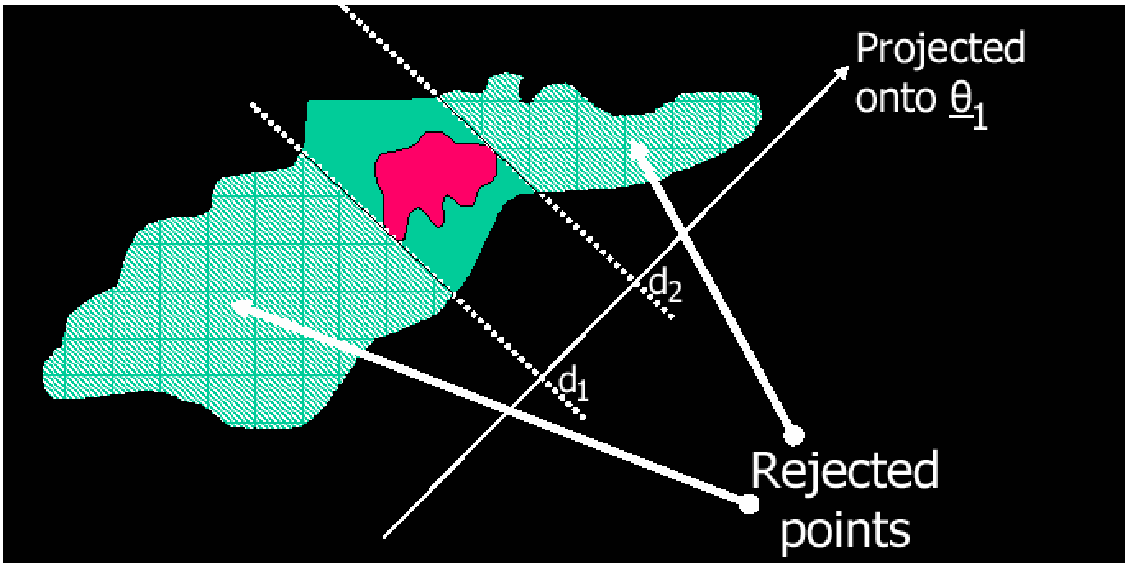


Figure 3.3: Maximal Rejection Classifier "1"

After that we take only the remaining non-face part in the dark green and finding theta two, it means that we must find another theta and two decision levels that the number of rejected non-faces is maximized while finding faces as shown in the figure below.

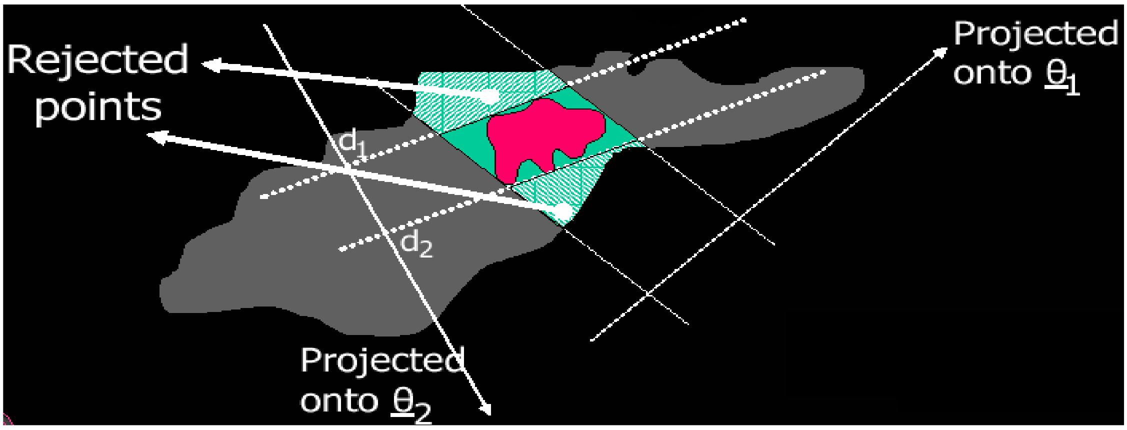


Figure 3.4: Maximal Rejection Classifier "2"

The first stage in the MRC is to gather two example-sets, faces and non-faces. Large enough sets are needed in order to guarantee good generalization for the faces and the non-faces that may be encountered in images.

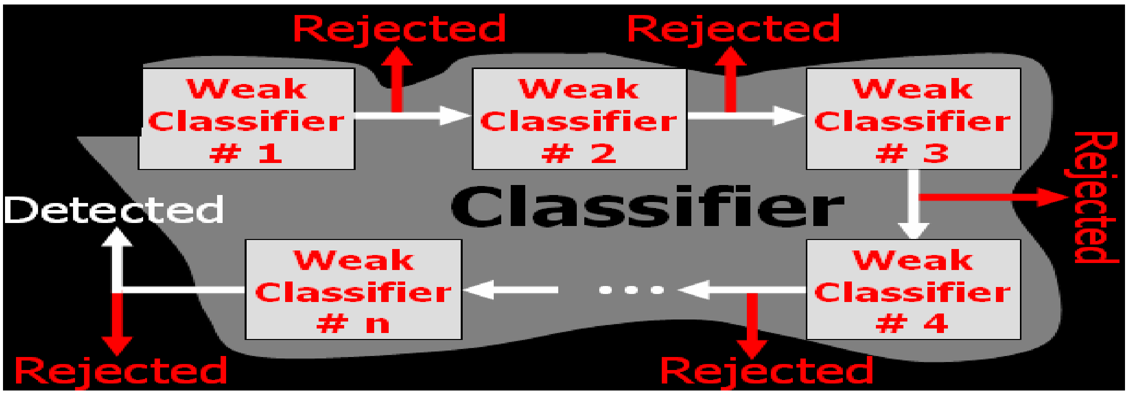
The target class should be assumed to be convex in order for the MRC to perform well. The target class containing frontal faces. One can easily imagine two faces that are not perfectly aligned, and therefore, when averaged, create a new block with possibly four eyes, or a nose and its echo, etc. we are using low- resolution representation of the faces (20 * 20 pixels).This implies that even for such misaligned faces ,the convex average appear as a face, and thus our assumption regarding convexity of the faces class is valid. The Non-face set is required to be much larger, in order to represent the variability of non-face patterns in images.

3.3.1 Rejection on MRC

The rejection in MRC is as follow:

- Build a combination of classifiers: we must find theta and two decision levels that the number of rejected non-faces is maximized while finding faces.

- Apply the weak classifiers sequentially while rejecting non-faces: it means that we must find another theta and two decision levels that the number of rejected non-faces is maximized while finding faces as shown in the figure below.



3.5: Maximal Rejection Classifier All Stages.

3.4. Multilayer Perceptron (MLP)

Multilayer perceptron networks belong to the class of supervised neural classifiers. They consist of perceptrons that are organized in layers: an input layer, one or more hidden layers, and the output layer. Every perceptron in one particular layer is usually connected to every perceptron in the layer above and below. These connections carry weights *wi*. Each perceptron calculates the sum of the weighted inputs, and feeds it into its activation function, regularly a sigmoid one. The result is then passed on to the next layer. The output layer has, for example, the same number of perceptrons as there are classes, and the perceptron with the highest activation will be consider the classification of the input sample. Training is achieved by successively feeding all training samples into the network, and comparing the output with the true class label.

The MLP can be categorized either as a fully connected, or as a partially connected network. A fully connected network is connected by all nodes between layers, unlike a partially connected network which is not connected at all nodes between former and next layers. A partially connected network can be used as a filter. The partially connected network uses less weights memory; however, it was proved that less connection reduced the performance through the simple experiment.

The detection rate of a fully connected network was better than a partially connected network. Moreover, it was very difficult to make the NN learn. Therefore, multi-layered and fully connected networks were used for the NN detector.

The MLP Network implemented for the purpose of this project as we see in figure 3.4 is composed of 3 layers, one input, one hidden and one output. The input layer constitutes of 400 neurons (20*20) which receive pixel binary data from a 20x20 symbol pixel matrix. The size of this matrix was decided taking into consideration the average height and width of character image that can be mapped without introducing any significant pixel noise.

The hidden layer constitutes of 300 neurons whose number is decided on the basis of optimal results on a trial and error basis. The output layer is 1 neuron.

To initialize the weights, a random function was used to assign an initial random number which lies between two preset integers named **weight_bias**. The weight bias is selected from trial and error observation to correspond to average weights for quick convergence.

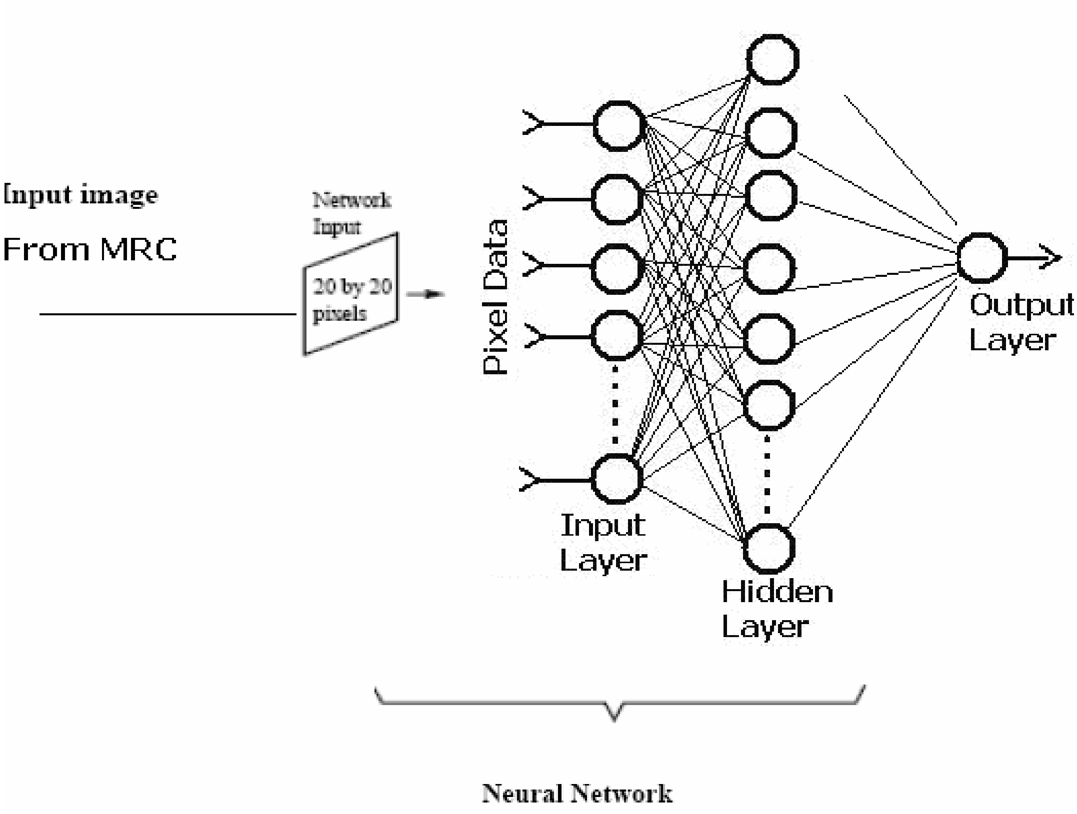


Fig.3. 6 the MLP Network

After the image data enters the input node, it is calculated by the weights. Face or non-face data is determined by comparing output results with the thresholds. For example, if the output is larger than the threshold, it is considered as a face data.

A problem that arises with window scanning techniques is overlapping detections. Deals with this problem through two heuristics:

- Thresholding: the number of detections in a small region surrounding the current location is counted, and if it is above a certain threshold, a face is present at this location.

- Overlap elimination: when a region is classified as a face according to thresholding, then overlapping detections are likely to be false positives and thus are rejected.

When we detect some face in the image the face is outlined with a bounding box and a copy of the image is displayed. In the next section a more thorough description of the system is included detailing the operation of the detector.

3.5 Summary

This paper introduces **a new face detection system** using cascading Neural Network and Maximal Rejection Classifier and is implemented using matlab and visual studio .NET). We will be dealing with approximately upright frontal face detection system of images in two-dimensional (2D). In neural network based face detection approach, the neural network examines an incremental small window of an image to decide if there is a face contained in each window as shown in figure 3.3. To decrease the amount of time needed for detection, the algorithm is enhanced by processing the image before it is fed to MRC and then to the network. This result in even better performance as probability of error is considerably reduced.

Figure 3.7: Small window of an image

One problem with MRC is that if given a very large input image, traversing it with 20x20 blocks at several resolutions could take a **long amount of time.** Another logistical problem lies in combining the results from multiple resolutions. If you find a face on one resolution, then find it again on another resolution, how to know if you found a new face or if it's the same one? This problem is actually not difficult to solve. The biggest problem with MRC is that it will obviously have some false detection. However, when we combined MRC with the NN finding algorithm we were able to severely cut down on these false detections.

In the face detection application, faces take the role of **targets** and non-faces are the **clutter**. In a typical image having millions of pixels, it is expected to detect a few of faces at the most, which means that picking a non-face block from the image is much more probable. This property is exploited by the MRC in order to obtain an efficient face detection classifier.

Chapter 4
Experimental Results and Discussion

In this chapter, we show an experimental result to presents the performance of our system. Firstly we compare the MRC alone to see its performance, secondly we compare the cascade system (MRC & MLPNN) to see its performance, and finally, we compare our system with other existing systems.

The face detection system presented in this paper was developed, trained, and tested using MATLAB™ 7.0, visual studio .NET 2003 on the Intel Pentium (4) 3.20 GHz 1.00GB of RAM and Windows XP operating system.
We divide our system into two parts; the first is training and the second is testing.

4.1 Training

Each stage in the cascade was trained using a positive set, and a negative set.
In order to train the Face Detection System to figure out the face and non-face images. We need to pass a collect (bunch) of face and non-face training data to give out 1 for face and -1 for non-face. Initially, our face training set contains face images collected from MIT face databases. Our face detection system has been applied to several test images (faces were frontal). All images were scaled to 20x20 pixels, and satisfactory results have been obtained. The test set consists of a total of 2000 face and non-face images given in the MIT database. Non-face patterns are generated at different locations and scales from images with various subjects, such as rocks, trees, buildings, and flowers, which contain no faces.

For the training images, because of the lighting, shadow, contract variances, we need to equalize those differences. The linear fit function will approximate the overall brightness of each part of the window.

To train the cascade of classifiers of the rejection stage to detect frontal upright faces, the same face and non-face patterns are used for all stages. The neural net is trained for 191 epochs as shown in figure 4.1.

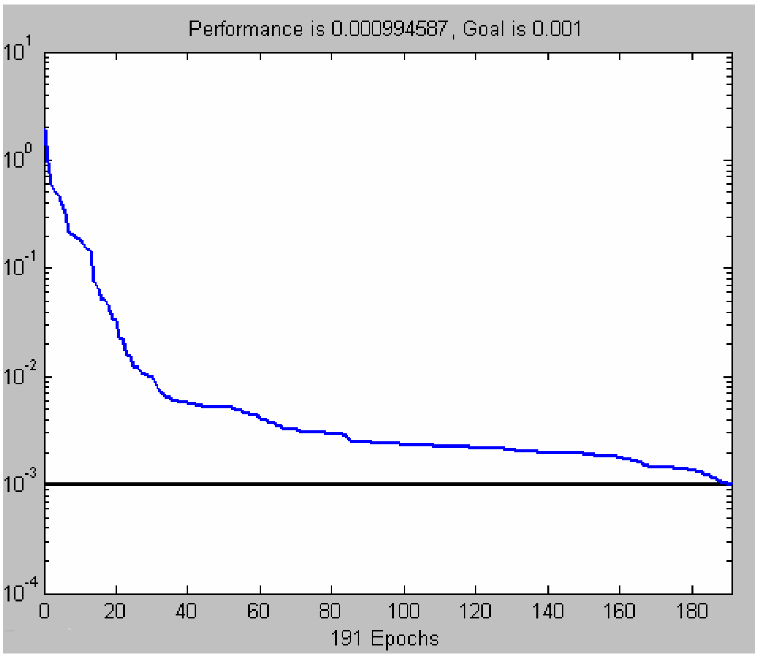


Figure 4.1: Trained Epochs.

4.2 System Testing

Firstly, we compare the first classifier learned using MRC algorithm on three images; secondly we compare those images on the all stages on our system. Finally, we compare our face detection system with others.

4.2.1 First Stage Testing

This experiment compares single strong classifier learned using MRC algorithm in the classification performance. The database consists of three images as shown in figure 4.2. The first image contains 8 frontal faces as we have shown in the picture below, the second picture contains 11 frontal faces, and the third picture contain just one frontal face. We test the first stage maximal rejection classifier **MRC** alone in our system to show the result of that stage .and then we test those images (Input images) on the cascade face detection system and show its result. The images below have complex background , occlusion faces and it has different shapes and different lighting.

Figure: 4.2 Test Images

Figure 4.3: Output of the first stage system

As we see in figure 4.3, the first stage can detect more than 90% of faces in all the images. And it has an average time 20 seconds to complete detect process. In the first image in figure 4.2 the first stage detect 92% , it means that it cannot detect occluded faces in the pictures. Second image in the same figure can detect all the faces without false negative, and with 34 seconds to complete the detected process. Table 4.1 shows the detection in Percent and time on the figure 4.2.

Table 4.1: Results in Fist Stage

Image #	Detection in Percent%	Avg. Time	False Negative	False Positive
1	92%	17 Sec.	None	2
2	100%	34 Sec.	None	None
3	100%	22 Sec.	None	None

4.2.2 Cascade Face Detection Testing (Second Stage)

This experiment shows our system result for both stages (MRC and MLPNN). The database consists of the same images as shown in figure 4.2.

Figure 4.4: Output of the cascade system

As we see the system has more performance to detect faces in all images with more accuracy and with acceptable average time. Table 4.2 shows the results.

Table 4.2: Results in the Cascade System

Image #	Detection in Percent%	Avg. Time	False Negative	False Positive
1	100%	19 Sec.	None	None
2	100%	42 Sec.	1	None
3	100%	26 Sec.	None	None

Figure 4.4 shows that the system can detect all the face in the first, second and third image with no false positive (misdetection) and with one false negative (false detection). But by comparing the system with first stage, the system increases the detection accuracy and rate for detection.

4.2.3 Comparison with other systems:

In this section comparison of the three algorithms is given. Table 4.3 shows the comparison of the algorithms. This comparison shows that our algorithm gives good results. Our system can be comparable with other, in mean of time and in detection rate. Table 4.3 shows the comparing results.

Table 4.3: Comparing with other systems

Classifier	Detection Rate	Error Rate
Multiple SVM Classifier	88.14%	5.86%
MLP Classifier	91.25%	8.75%
adaboost Classifier	99.78 %	0.22 %
Our system (MRC & MLP Neural Network)	91.6%	7.54%

The table above shows that our system has detection rate more than SVM and MLP algorithms and our system has an error rate (False positive and false negative) less than Ad-boost classifier.

4.3 Face Database

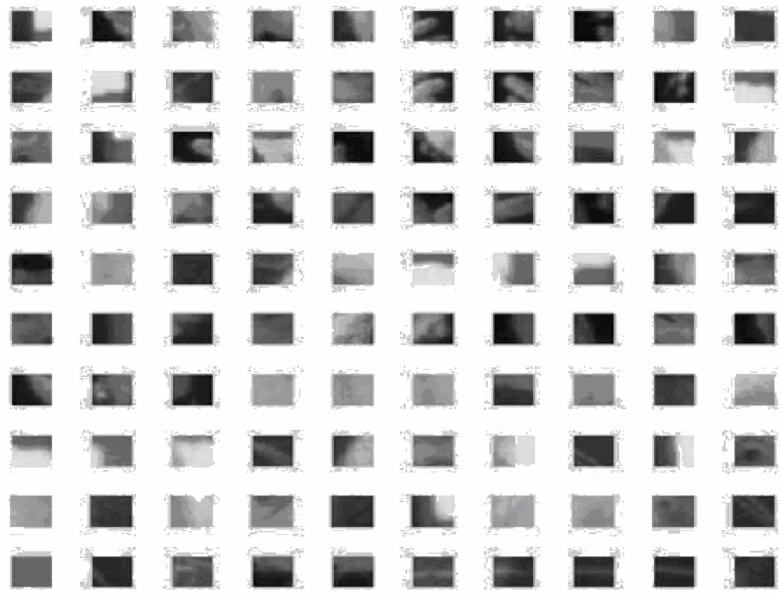
Most face detection methods require a training data set of face images and the databases (as shown in table 4.1) originally developed for face recognition experiments used as training sets for face detection [35].

Table 4.4: Face Data Base

Data Set	Location
MIT Media Labs Database	ftp://whitechapel.media.mit.edu/pub/images/
MIT CBCL Face Data Set	http://www.ai.mit.edu/projects/cbcl/software-datasets/FaceData2.html
FERET Database	http://www.nist.gov/srd/
UMIST	http://images.ee.umist.ac.uk/danny/database.html
University of Bern	ftp://iamftp.unibe.ch/pub/Images/FaceImages/
Yale Database	http://cvc.yale.edu
AT&T (Olivettti) Database	http://www.uk.research.att.com
Harvard Database	http://cvc.yale.edu/people/faculty/belhumeur.html
M2VTS Database	http://poseidon.csd.auth.gr/M2VTS/index.html
Purdue Database	http://rvl1.ecn.purdue.edu/~aleix/aleix_face

Our face detection system has been applied to several test images (faces were frontal) as shown in figure 4.1. All images were scaled to 20x20 pixels, and satisfactory results have been obtained. The test set consists of a total of 2000 face and non-face images given in the MIT database. Non-face patterns are generated at different locations and scales from images with various subjects, such as rocks, trees, buildings, and flowers, which contain no faces.

4.5: Face Examples from Face training set.



4.6: Non-Face Examples from Non-Face training set.

4.4 Performance Analysis

To evaluate the performance of our face detection system, we will execute our system with a test set containing above 2 databases of digital photos of people from internet. Then we will compare our result with other face detection algorithms. A related and important problem is how to evaluate the performance of the face detection methods [1]. Many recent face detection papers compare the performance of several methods, usually in terms of detection and false alarm rates. It is also worth noticing that many metrics have been adopted to evaluate algorithms, such as learning time, execution time, the number of samples required in training and the ratio between detection rates and false alarms. In general, detectors can make two types of errors: ***false negatives*** in which faces are missed resulting in low detection rates and ***false positives*** in which an image is declared to be face.

To evaluate the performance of our face detection method, we executed the algorithm for a test set containing 30 downloaded digital photos of people (most of them from *Yahoo* photo gallery).

Now that the system is defined, we can proceed to analyze its performance. In the following table we have run our face detection system with a number of images and thresholds and tallied the correct face identification rate and the false positive rates over all images. The number for the false positive rate is the number of false positives out of all rectangles classified for the set of images. One can easily see the trade off between correct face identification rate and false positive rate as a function of the threshold as mentioned above.

Table 4.5: Thresholds versus Detection Rate

Threshold	Correct Face ID Rate	False Positive Rate
0.0	1.00	35/6500
0.1	1.00	21/6500
0.2	1.00	15/6500
0.3	0.857	7/6500
0.4	0.714	6/6500
0.5	0.714	4/6500
0.6	0.571	3/6500
0.7	0.429	1/6500
0.8	0.286	0/6500
0.9	0.000	0/6500

Table 4.6: Detection Rate Versus Size of Pixel

	Size = 100 (10 × 10)	Size = 150 (15 × 10)	Size = 400 (20 × 20)
Detection Rate (%)	77.72	80.5	91.2

We choose 20* 20 size of pixel because it gives a maximal detection rate as we shown in Table 4.6 above.

Our algorithm can detect between 80% and 92% of faces in a set of 135 test images, with an acceptable number of false detections. Depending on the application, the system can be made more or less conservative by varying the arbitration heuristics or thresholds used. The system has been tested on a wide variety of images, with many faces and unconstrained backgrounds. A fast version of the system can process a 320x240 pixel image in 2 to 4 seconds on a 3.2 GHz pentium4 with 1024 Megabyte Ram.

There is a number of directions for future work. The main limitation of the current system is that it only detects upright faces looking at the camera.

4.5 Summary

The algorithm presented in this thesis can detect between 80% and 92% of faces in a set of 135 test images with complex backgrounds and lighting, with an acceptable number of false detections. The system has been tested on a wide variety of images. And we have developed an algorithm for face detection based primarily on rejection techniques. Rejection is a powerful method and it increases the performance in any field. The maximal rejection classification technique proved to be effective even though it is a linear technique. Thus, we have taken advantage of the speed of a linear algorithm while still maintaining a high accuracy.

Chapter 5
Conclusion

In this paper we presented a new classifier for face detection, our Algorithm can detect between 79% and 92% of faces from images of varying size, background and quality with an acceptable number of false detections. A threshold of between 0.5-0.6 gives the best range of results out of the threshold set tested. The new system was designed to detect upright frontal faces in color images with simple or complex background. There is no required a priori knowledge of the number of faces or the size of the faces to be able to detect the faces in a given image. The system has acceptable results regarding the detection rate, false positives and average time needed to detect a face.

The results indicate that our system can be more suitable in the given situation when compared with a standard multi-layer perceptron.

We observe that the implemented method for face detection is quite successful given a dataset of images. Actually we have taken a subset of the available database. If the number of images in the dataset increases, the probability of getting better results increases. As we have observed from our experiments the threshold values chosen also affect the output considerably.

The advantage of our algorithm is that it can be used to track any object instead of human face and the obtained detection result is shown to be both very efficient and accurate. This, after, all was the goal of the system. The main limitation of the current system is that it only detects upright frontal faces.

5.1 Future Work

There is a number of directions for future work. One of the assumptions of our system was that the face in an input image should not be rotated, this constraint which can be overcome if we include second neural network for rotations. Finally, the proposed method could, for example, serve as a first part of a face recognition system. Falsely, detected faces could be eliminated using correspondence to the database of already known faces.

References

[1] Aamer Mohamed, Ying Weng, Jianmin Jiang and Stan Ipson," Face Detection based Neural Networks using Robust Skin Color Segmentation", Location:Amman,ISBN:978-1-4244-2205-0, IEEE SSD 2008. 5th International Multi-Conference, 20-22 July 2008 Page(s):1 – 5.

[2] Aamer .S.S.Mohamed, Ying Weng, Stan S Ipson, Jianmin Jiang," Face Detection based on Skin Color in Image by Neural Networks", Accepted by IEEE International Conference on Intelligent and Advanced Systems ICISA 2007

[3] Abdallah S. Abdallah, Mohamad A bou El-Nasr, and A. Lynn Abbott," A New Color Image Database for Benchmarking of Automatic Face Detection and Human Skin Segmentation Techniques",PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 20 APRIL 2007

[4] A. M. Martinez and R. Benavente, “The AR face database,” tech. rep., CVC #24, 1998.

[5] Aouatif Amine, Sanaa Ghouzali and Mohammed Rziza ,” Face Detection in Still Color Images Using Skin Color Information”, ISBN: 978-1-4244-1751-3 ,ieee, April 2008.

[6] Asim Shankar, Priyendra Singh Deshwal,” Face Detection in images : Neural networks & Support Vector Machines ”, April 2002, Indian Institute of Technology, Kanpur.

[7] B. Heisele, T. Poggio, and M. Pontil, “Face detection in still gray images,” Tech. Rep. 1687,Massachusetts Institute of Technology, Cambridge, Mass, USA, 2000, AI Memo.

[8] Chiunhsiun Lin, "Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network", Volume 28 , Issue 16 (December 2007), Pages 2190-2200, acm.
[9] Christophe Garcia and Manolis Delakis," A Neural Architecture for Fast and Robust Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26 , Issue 11 (November 2004).

[10] Catalin-Daniel, Corina Botoca," C# Solutions for a Face Detection and Recognition System", FACTA UNIVERSITATIS ,vol. 20, no. 1, April 2007.

[11] C. Caleanu, "Facial recognition using committee of neural networks," in 5th Seminar on Neural Network Applications in Electrical Engineering (NEUREL-2000), Belgrade, Yugoslavia, 2000.

[12] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1997, pp. 130–136.

[13] Feraund, R.; Bernier, O.J.; Viallet, J.-E.; Collobert, M. "A fast and accurate face detector based on neural networks", IEEE, Volume 23, Issue 1, Jan 2001 Page(s):42 – 53.

[14] Feraund, R.; Bernier, O.J.; Viallet, J.-E.; Collobert, M. "A Face Detection System using Neural Network Approach", IEEE, VOL. 23, NO. 1, JANUARY 2001.

[15] FA˙IK BORAY TEK," FACE DETECTION USING LEARNING NETWORKS", www.eee.metu.edu.tr/~alatan/PAPER/MSboray.pdf, 2002.

[16] F. Zuo and P. H. N. de With, "Fast face detection using a cascade

of neural network ensembles," in Proceedings of the 7th International Conference on Advanced Concepts for Intelligent Vision , vol. 3708 of LNCS, pp. 26–34, September 2005.

[17] Fei Zuo1 and Peter H. N, " Cascaded Face Detection Using Neural Network Ensembles", 8 October 2007, Eindhoven University of Technolog.

[18] F. Smach, M. Atri, J. Mitéran and M. Abid, "Design of a Neural Networks Classifier for Face Detection",Journal of Computer Science , march, 2006.

[19] Fei Zuoa and Peter H.N. de Fast Human Face Detection Using Successive Face Detectors with Incremental Detection Capability

[20] G. Yang and T. S. Huang, "Human face detection in complex background," Pattern Recognition, vol. 27, no. 1, pp. 53–63, 1994.

[21] Hazem M. El-Bakry and Qiangfu Zhao," Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform",International Journal of Signal Processing, www.waset.org Fall 2005.

[22] Hazem M. El-Bakry and Qiangfu Zhao," Sub-Image Detection Using Fast Neural Processors and Image Decomposition",PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 4 FEBRUARY 2005 .

[23] Hazem M. El-Bakry," A New Implementation of PCA for Fast Face Detection", International Journal of Intelligent Systems and Technologies, www.waset.org Spring 2006

[24] Hazem M. El-Bakry, "Fast Neural Networks for Pattern Detection Using 2D-FFT"

[25] H. Zhang, B. Zhang,W. Huang, and Q. Tian, “Gabor wavelet associative memory for face recognition,” IEEE Trans. Neural Networks, vol. 16, no. 1, p. 275278, Jan. 2005.

[26] H. Martin Hunke, ”Locating and tracking of human faces with neural networks”, Master’s thesis, University of Karlsruhe, 1994.

[27] I. Lawrence, C. Giles, A. Tsoi, and A. Bock, “Face recognition: A convolutional neural-network approach,” IEEE Trans. Neural Networks, vol. 8, pp. 98–113, 1997.

[28] James Cannady,” Artificial Neural Networks for Misuse Detection”

[29] Kevin Curran, Xuelong Li and Neil Mc Caughley," The Use of Neural Networks in Real-time Face Detection", Journal of Computer Sciences, 47-62, 2005.

[30] K.A. Ishak, S.A. Samad, A. Hussain and B.Y. Majlis,"A FAST AND ROBUST FACE DETECTION USING NEURAL NETWORKS",

[31] KALLEL, R.—COTTRELL, M.—VIGNERON, V. : Bootstrap for Neural Model Selection, in Proc. of ESANN’2000 – European Symposium on Artificial Neural Networks, Bruges (Belgium), 26-28 April 2000, pp. 61–68.

[32] Lamiaa Mostafa and Sherif Abdelazeem,"Face Detection Based on Skin Color Using Neural Networks",GVIP 05 Conference, 19-21 December 2005, CICC, Cairo, Egypt.

[33] Lin-Lin Huang, Akinobu Shimizu, Yoshihiro Hagihara, Hidefumi Kobatake, "FACE DETECTION FROM CLUTTERED IMAGES USING A POLYNOMIAL NEURAL NETWORK"

[34] Mohammad Inayatullah, Shair Akbar Khan ,Bashir Ahmad. “A Face Detection System using Neural Network Approach”, IC-AI 2006: 753-758.

[35] Ming-Hsuan Yang Kriegman, D.J. Ahuja, N.," Detecting faces in images: a survey", Jan 2002, page(s): 34-58, vol 24, IEEE.

[36] Mohabbati, B.; Kasaei, S.," An efficient wavelet/neural networks-based face detection algorithm", The First IEEE and IFIP International Conference in Central Asia on Volume , Issue , Publication Date: 26-29 Sept. 2005 , On page(s): 5 ,ISBN: 0-7803-9179-9.

[37] Marian, Milos," A SYSTEM FOR LOCALIZATION OF HUMAN FACES IN IMAGES USING NEURAL NETWORKS", Journal of ELECTRICAL ENGINEERING, VOL. 56, NO. 7-8, 2005.

[38] M. Kunze and J. Steffens, "The neural network objects," in Proceedings of the fifth IHENP Workshop, Lausanne, 1996.

[39] Phil Brimblecombe,"Face Detection using Neural Networks",

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

[41] Rowley, H.A., S. Baluja and T. Kanade," Neural network-based face detection" . 1998 ,IEEE.

[42] Rowley, S. Baluja, and T. Kanade, "Rotation Invariant Neural Network-Based Face Detection", Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 38-44, 1998.

[43] R. L. Hsu and M. A.-M. an A. K. Jain, "Face detection in color images," IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 696–706, 2002.

[44] S. Ben-Yacoub, "Fast Object Detection Using MLP and FFT", technical report, IDIAP, Martigny, Switzerland, 1997…………"put it in PDF"……………………...

[45] S Wiegand, C Igel, U Handmann," Evolutionary optimization of neural networks for face detection",12th European Symposium on Artificial Neural Networks (ESANN 2004).

[46] S. Lin, S. Kung, and L. Lin, "Face recognition/detection by probabilistic decisionbased neural networks," IEEE Trans. Neural Networks, vol. 8, pp. 113–123, 1997.

[47]" A face detector using Neural Networks and Discrete Wavelet Transforms ",Lnes, Boaventura ,Valeria M Volpe.

[48] Example-based learning for Human Face Detection. Kah-Kay Sung, Tomaso Poggio. MIT. IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20, number 1, pages 39-51, January 1998.

[49] Pattern detection using a maximal rejection classifier, Elad, M.; Hel-Or, Y.; Keshet, R. Volume , Issue , 2000 Page(s):193 – 194.