INVESTIGATION OF NEW TECHNIQUES

FOR FACE DETECTION

Abdallah S. Abdallah

Thesis Submitted to the Faculty of
Virginia Polytechnic Institute and State University
In Partial Fulfilment of the Requirements of the degree of

Master of Science
in
Computer Engineering

A. Lynn Abbott, Chairman
M. Abou El-Nasr, Co-Chair
Leyla Nazhand Ali
Peter Athanas

May 9, 2007
Blacksburg, Virginia

Keywords: Face Detection, Skin Segmentation, Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Edge Detection, Geometrical Moments, Fusion, Hybrid Feature Vector, Self Organized Map (SOM)

INVESTIGATION OF NEW TECHNIQUES FOR FACE DETECTION

Abdallah S. Abdallah

ABSTRACT

The task of detecting human faces within either a still image or a video frame is one of the most popular object detection problems. For the last twenty years researchers have shown great interest in this problem because it is an essential pre-processing stage for computing systems that process human faces as input data. Example applications include face recognition systems, vision systems for autonomous robots, human computer interaction systems (HCI), surveillance systems, biometric based authentication systems, video transmission and video compression systems, and content based image retrieval systems.

In this thesis, non-traditional methods are investigated for detecting human faces within color images or video frames. The attempted methods are chosen such that the required computing power and memory consumption are adequate for real-time hardware implementation. First, a standard color image database is introduced in order to accomplish fair evaluation and benchmarking of face detection and skin segmentation approaches. Next, a new pre-processing scheme based on skin segmentation is presented to prepare the input image for feature extraction. The presented pre-processing scheme requires relatively low computing power and memory needs. Then, several feature extraction techniques are evaluated. This thesis introduces feature extraction based on Two Dimensional Discrete Cosine Transform (2D-DCT), Two Dimensional Discrete Wavelet Transform (2D-DWT), geometrical moment invariants, and edge detection. It also attempts to construct a hybrid feature vector by the fusion between 2D-DCT coefficients and edge information, as well as the fusion between 2D-DWT coefficients and geometrical moments. A self organizing map (SOM) based classifier is used within all the experiments to distinguish between facial and non-facial samples. Two strategies are tried to make the final decision from the output of a single SOM or multiple SOM. Finally, an FPGA based framework that implements the presented techniques, is presented as well as a partial implementation.

Every presented technique has been evaluated consistently using the same dataset. The experiments show very promising results. The highest detection rate of 89.2% was obtained when using a fusion between DCT coefficients and edge information to construct the feature vector. A second highest rate of 88.7% was achieved by using a fusion between DWT coefficients and geometrical moments. Finally, a third highest rate of 85.2% was obtained by calculating the moments of edges.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the closest person to my heart who I wish I was able to spend every moment in my life with her since the day I was born. I struggled all the way motivated by the idea that my wishes will become true one day. Her love, care, and support make me believe that nothing is impossible.

I joined the Virginia Tech Middle East and North Africa (VT-MENA) program for graduate studies in August 2005, since then I had learnt a lot of amazing concepts of science and life. VT-MENA professors gave me the required inspiration and the spirit of challenge to take the first steps in my career as a teaching assistant and a graduate researcher in the computer engineering field. I will mention some examples here and I hope if I have enough space to mention everything.

First of all, great thanks to my advisor Dr. A. Lynn Abbott, and my co-advisor Dr. M. Abou El- nasr for guiding me step by step through the maze of research in computer vision and image processing. A great part of the knowledge I got relates to the continuous encouragement and assistance they gave me. They succeeded to inspire me to wok harder and aim higher all the time.

Thanks to Prof. Yasser Hanafy the MENA resident project director, for his precious advice to join the VT-MENA program. I really appreciate his great efforts and amazing support making all the facilities available for me and other VT-MENA students.

Thanks to Dr. Sedki Riad the US based VT-MENA director, for facilitating any required admission, registration or paper work in Blacksburg. He is really so kind and patient person. Special thanks go to Cindy Hopkins, the one that I never met but I know that she is always there to offer help or advice.

Thanks to VT-MENA visiting professors who came to Egypt and shared their knowledge and culture with us, especially Dr. A. A. Louis Beex.

Thanks to VT-MENA adjunct faculty members especially Dr. Ayman Adel. I can’t imagine meeting someone else who is so dedicated to his work and his students like the way he is.

Thanks to Dr. Leyla Nazhand Ali and Dr. Peter Athanas for accepting to be a part of my committee. Special thanks go to Dr. Athanas for providing the required software license to accomplish this research.

I can’t thank my parents enough for their love and support. Thanks also go to my sisters and brothers for their encouragement. I also would like to thank my special aunts who have been advising and guiding me through my college study years.

Abdallah Sabry Abdallah
April 10, 2007

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Problem Definition

Face detection can be defined as the computer based process that takes an image as input and produces a set of image coordinates where human faces are located if present. Comprehensive surveys on the area are given in [1] and [2]. Figure 1.1 presents an example for detecting of human faces in an input image.

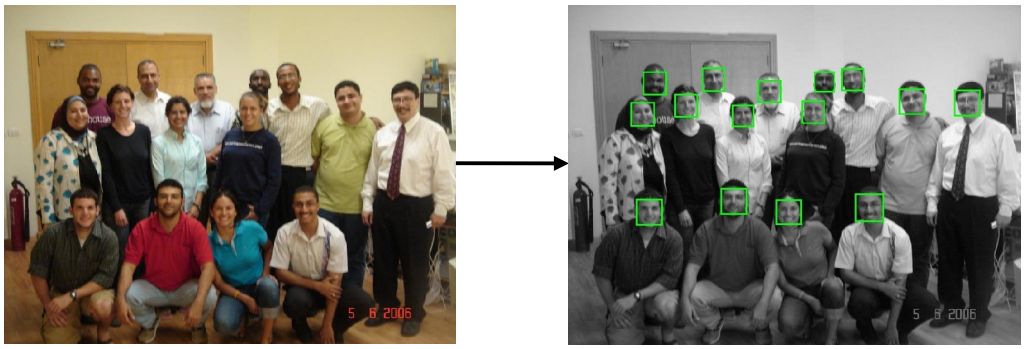


Figure 1.1: An example of face detection. The faces have been manually detected for illustration purposes.

The face detection process is a basic pre-processing stage for any computer-based system that processes images or video streams that deal with the human face. Example applications include face recognition, surveillance, face tracking, human-computer interaction (HCI), robotic vision and autonomous vehicles, biometric based authentication, content based image retrieval, as well as selective compression.

The great challenge for the face detection problem is the large number of factors that govern the problem space. The long list of these factors include the pose, orientation, facial expressions, facial sizes found in the image, luminance conditions, occlusion, structural components, gender, human race, the scene and complexity of image's background. Although a large number of face detection techniques and methods have been presented in the literature, few approaches address more than one or two variables. They assumed their own problem constraints to limit the problem space instead of challenging this wide range of factors.

1.2 Motivation

The basic goal of this thesis has been to come up with a new face detection scheme that satisfies the need for real-time hardware implementation. These criteria are mainly the required computing power and storage space. In order to accomplish this goal, we selected a preprocessing method that depends on skin segmentation, morphological opening, and labeling of connected regions to find the candidate skin regions that are most likely to be faces. We also decided to investigate the usage of a set feature extraction methods which includes the discrete cosine transform (DCT), the discrete wavelet transform (DWT), Hu moment invariants, and the Canny

algorithm for edge detection. All of these methods have the advantage that they are all implement-able in real-time hardware.

Transformation based image analysis techniques, such as the two dimensional discrete cosine transform (2D-DCT) and the discrete wavelet transform (2D-DWT), have been widely used in the field of image compression. These form the foundation for the compression image algorithms like JPEG [39] and JPEG2000 [43]. The ability of the DCT and DWT to maintain the information contained in an image in a compressed form motivates their use for feature extraction. Each DCT or DWT coefficient can be considered as an independent variable in a feature space. The correlation between the values of the different coefficients among different patterns of various classes is used to construct classification boundaries. The DCT has been used for recognition of facial expressions [40, 41], and this suggests that it may give good results for face detection. The small number of coefficients of the DWT promotes it to be used to save computational power and memory consumption. The main advantage of the DCT and DWT is that they are feasible for hardware implementation to perform real time face detection. Both have been implemented for many different applications [72, 73, 74, 75, 76, 77].

The real challenge of using the geometrical moments for feature extraction is the heavy required computing power and memory consumption. The collection of Hu moment invariants has a significant advantage over other geometrical moments because it is the simplest to be calculated [50]. The invariance property of Hu moments against geometrical transformations like scaling, translation, and rotation makes it a good candidate feature extractor to be used for face detection. It is expected to have no effect on its result due to diversity in facial pose, two-dimensional orientation, or size. In addition, there were few attempts to use the geometrical moments for face detection; the attempt whose results were published in [71] is one of these trials. All of the above encourages to reinvestigating the use of geometrical moments in face detection. Also, edge detection methods were not used for a long time although they are very traditional powerful tools for feature extraction. Because of that, we plan to reinvestigate the use of edge detection both alone and combined with the previously mentioned methods seeking higher detection rates.

In addition, there is another motivational issue related to the comparative evaluation of the large set of face detection approaches that were presented in the literature. The issue that was addressed in [1, 2] is the ambiguity about some of the published results of previous approaches due to the lack of a standard evaluation procedure. In particular there is no standard database for testing and evaluating face detection techniques. The CMU database is a widely used database since Rowley [3, 4] but it is only useful for detecting faces in grayscale images. This motivated us to compile a new color face image database called the VT-AAST database that is presented in chapter three. It is a color image database for the testing and evaluation of face detection and skin segmentation techniques. It includes a realistic set of test cases that pose a real challenge to any new proposed approach.

Since the final goal is to implement the investigated methods in hardware, an FPGA based framework for real-time face detection is proposed. Also a small part is implemented as a case of study to address some issues that are related to the hardware implementation.

1.3 Approach

The aim of this thesis is to investigate the performance of a novel face detection system that has been developed specifically for hardware implementation. The novel scheme includes three stages. These stages are summarized as follows:

1. Pre-processing
2. Feature extraction
3. Learning based classification

The first stage uses skin segmentation, morphological opening, and region labeling for preprocessing the input image to extract the candidate skin regions that are likely to represent faces. The second stage extracts the features of the candidate regions. Several experiments have been done to investigate the performance of four feature extraction methods including DCT, DWT, Hu moment invariants, and the Canny edge detection. The last stage uses a self organizing map neural network for classifying the candidate regions as either face or non-face.

The aim is not only about achieving a high detection rate. In addition to good performance, the computational complexity of the proposed scheme has to be practical in terms of processing time and memory requirements. To satisfy such criteria, the statistical principal component analysis is used to select a subset of the features space to construct the feature vector. This limits the size of the feature vector to meet the capabilities of the computational power and storage space that are offered by the implementation technology whether the system is implemented in software or hardware.

1.4 Significance of Research

The research has focused mainly on investigating the performance of the proposed feature extraction techniques using the self organizing map as learning based classifier. It has also demonstrated the feasibility of implementing a complete real-time face detection system based on the proposed techniques. This research has accomplished several achievements that can be summarized as follows:

- This research has introduced a standard color image database for testing, and benchmarking of face detection and skin segmentation techniques. It is called the VT-AAST face detection database.

- To the best of our knowledge, this thesis is one of the few attempts to investigate the usage of either the discrete cosine transform or the Hu moment invariants in the feature extraction process for face detection.

- Two schemes have been presented to generate a hybrid feature vector which is new idea to the face detection. The first scheme combines the DCT coefficients with the edge information where the second uses the fusion between DWT coefficients and Hu moment invariants to generate the hybrid feature vector. The effect of reducing the dimensionality of the hybrid feature vector in order to select the most un-correlated features is also investigated.

- A novel voting scheme to combine the decisions of multiple parallel classifiers is introduced in order to increase the probability of correct decisions.

- All the proposed methods and techniques are implement-able using real-time hardware.

- Evaluation of the detection parameters and performance measures of the proposed techniques provides very promising results comparable to the previous work. The comparison is interesting because of the high level of difficulty of the test cases given by the VT-AAST database, as compared to the previous benchmarking datasets.

- A framework for an FPGA based implementation is proposed. This is considered a good step toward final implementation of a real-time face detection system. A partial implementation that covers the preprocessing stage to the end of the skin segmentation process is also presented. This is a suggested direction for expanding this research in future work.

1.5 Thesis Organization

This thesis is organized as follows. Chapter one gives an introduction to the field of face detection. It presents the problem definition and variables, the motivation behind the presented research, and the significance statement of this research.

Chapter two provides a brief summary of the previous face detection research. It demonstrates the taxonomy that organizes the early research attempts. It also presents the significant achievements related to the face detection. In particular, the most recent accomplishments whether they are based on offline processing or real-time processing.

Chapter three presents a detailed survey of the existing datasets that were used prior to this research for the testing of face detection methods and techniques. Then, a new standard color image database is presented for the benchmarking of face detection and skin segmentation techniques. This is the database that was used to test and evaluate the presented techniques in this thesis.

Chapter four provides the reader with a full theoretical background that is needed to understand most of the fundamentals of image processing and analysis methods that are involved in this thesis.

Chapter five provides details of the stages and procedures that were followed during the experiments, as well as the experimental results and interpolation for every experiment. It also provides a final comparison between the results, giving reasoning for the differences among them.

Chapter six considers the hardware implementation of the presented face detection scheme in this thesis. It suggests an FPGA based framework for a hardware implementation. A system block diagram for the hardware framework is also presented. In addition a partial implementation for a small part of the system is presented as a case study for the hardware implementation. Finally, Chapter seven will conclude the work and discuss directions for future work.

CHAPTER 2
Background and Related Work

2.1 Background

The performance of the face detection operation may be highly affected by numerous factors. Each factor is related to a real life attribute that may describe a human face found in the input image, the input image itself or the environmental conditions at the moment when the image has been taken. This section describes most important factors as well as the expected effects of their existence on the performance of the face detection process.

Three dimensional pose refers to the degree of face rotation around the vertical axis of the input image. Poses can be categorized into frontal, for un-rotated faces, profile, for faces rotated approximately 90 degrees, and intermediate, for faces in between. It may cause partial or complete occlusion of facial features, including the mouth, nose, or eyes. *Two dimensional orientation* refers to the degree of face rotation around the axis that is perpendicular to the image plane. The faces with a very small rotation range are called upright faces and the rotation may increase clockwise or counter clockwise until the face becomes upside down.

There is also the effect of the divergence of *facial expressions*. The detection and analysis facial expressions and gestures is an independent field of research by itself. Various techniques and approaches were presented for facial expression analysis [70, 40, 41]. To the best of our knowledge, no one addressed this criterion within face detection, although the fact that the appearance of facial features in an image snapshot is highly influenced by the apparent facial expression at the moment.

Another effective factor is the *variation of facial sizes* in the image. The dynamic range of face sizes even in a single image is inevitable so that the face detection has to be done at different scales. This effect was handled by other researchers by using image multi-resolution analysis along with the assumption of a minimum size of the face to limit the search space [3, 4]. Another alternative is to use a lower level feature like skin color to localize the candidate regions that might be faces in order to also limit the search space.

The range of luminance in the image is one more crucial factor. It is a relevant feature to the lighting conditions and camera setup. Several techniques were used to eliminate the effect of luminance variation [8, 14, 16]. For face detection within a color image, normalized color space can be used to eliminate the drastic change in intensity values before converting the image into grayscale. Another alternative is to consider only the chrominance components while discarding the luminance values.

There is also but not only the occlusion effect which is the partial blocking of some facial features by another object such as another face or any other object from the surrounding environment, as well as the *scene and background* complexity which announces constraints on the nature and complexity of background and surrounding environment to eliminate the effect of the possible variation of them. Indoor and outdoor scenery, as well as simple and cluttered backgrounds present a great dynamic

and non-predictable challenge for any face detector. The most obvious effect is the existence of *structural components* like facial hair, eyeglasses, and sunglasses. These particular objects may partially cover the face causing the miss of some facial features during the detection process. Finally, the variance of the gender or human race has also its effect on the extraction of facial feature. The studies have shown that the difference between the skin colors of different races has an effect primarily on the luminance not the chrominance [2] so that the skin tone of people from different races falls within the same range of chrominance. This thesis takes into account the race definitions given in [18].

2.2 Previous work

This section presents a taxonomy that categorizes most of the well known face detection algorithms and techniques. It also presents a summary of the most significant recent research related to face detection. For a comprehensive survey of the related face detection research work that were accomplished prior to May 2000, check Yang et al. [1] as well as Eric and Low [2].

2.2.1 Overview

The most well-known face detection work is the work done by Rowley [3, 4]. This is because that work is the first well documented and extensively evaluated research in the field. They first presented an upright frontal face detector [3] before extending their work to include detection of rotated faces [4]. Both systems used neural network based classifiers.

The previously published methods for face detection were classified into two major categories [1, 2]. Erik and Low [2] presented a more detailed classification than Yang et al. in [1], although they are pretty similar in general. The following is a modified taxonomy that is also summarized in figure 2.1.

The first category assembles what are called rule based methods. They are the methods that rely on searching the input image comprehensively for facial features under a predefined set of rules. This category could be partitioned by itself into two sub categories based on the nature of the rules in use. The rules are either knowledge based rules or feature invariant rules. For the knowledge based methods, the rules that govern the search are arbitrary inherited from the human knowledge of typical faces. Yang et al. [1] stated that this category was developed mainly for face localization, not for face detection. However the feature invariant based methods rely on using a combination of one or more low level features such as edges, gray levels, skin color, texture, or even motion if the input is a video stream. This is done in order to search the input image for a single facial feature or multiple facial features (nose, mouth, or eyebrows) to detect the target face. Methods belong to the rule based category are rather more adequate for face localization, not face detection.

As for the second category, the methods belong to it are called model based methods. They are classified into two sub categories. First, the template matching based methods where a model is pre-specified for a human face includes a predefined structure for the different facial features and their inner-relations. The model is either fixed or parameterized to adapt to the dynamic changes in scale, pose, and shape. The

correlation between the input image and the model is measured for several facial features independently and then a decision is made based on the measured correlation. Examples of the fixed templates can be found in [1]. Examples of parameterized templates that are also known as active shape models include snakes, deformable templates, and point distributed models. Second, Image based methods where a model is learnt of examples from the training samples set. The number of training samples is important because it must be enough to collect as much facial features as possible using one or more feature extraction methods. The classification is done using a specific machine learning technique. Examples of this category include a list of statistical based methods and machine learning based techniques such as eigenfaces, distribution-based methods, support vector machines (SVM), artificial neural networks (ANN), sparse network of Winnows (SNoW), naïve Baye's classifiers, and hidden Markov model (HMM).

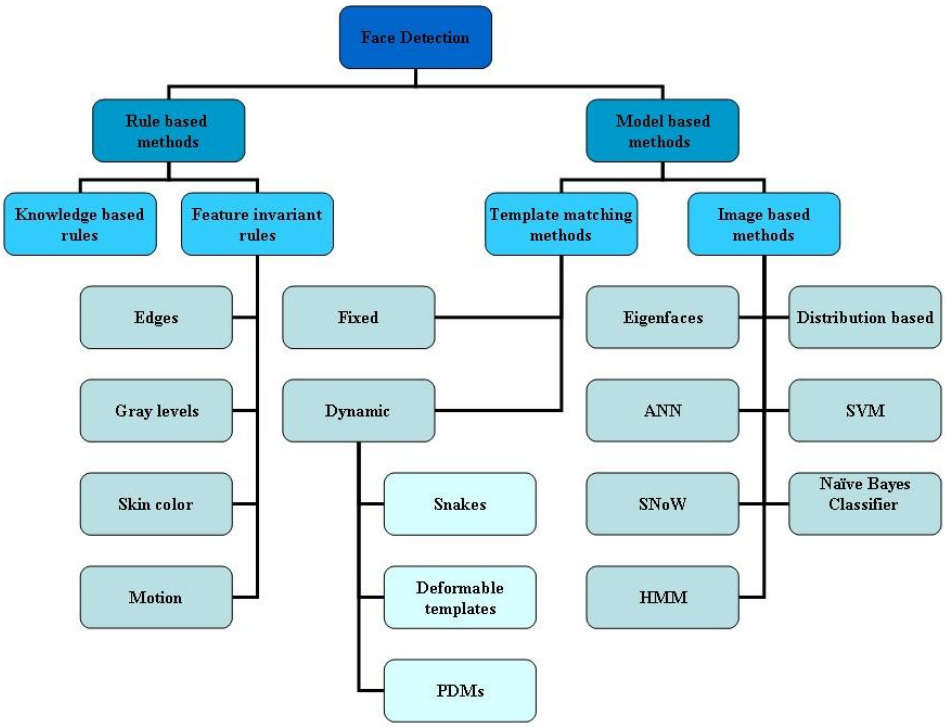


Figure 2.1: Taxonomy of face detection techniques.

2.2.2 Recent related work

This section summarizes some of the recent related work that was published after those comprehensive surveys [1, 2]. All the following are learning based techniques that may be classified into two categories. The first category includes techniques that were developed in software for processing on regular machines. The second category includes techniques that were developed for real time running on embedded systems.

First, face detection systems that have been developed to operate as an offline or real-time software based detectors. Some of these systems are summarized below: In [81] Sandeep and Rajagopalan proposed a fast algorithm for face detection within color images using a color histogram for human skin in HSV space combined with edge information. No standard dataset or well-known dataset was used but the histogram analysis was done using over 450,000 pixels drawn from different internet sources, so that they belong to people of different races. Specific detection rates were not mentioned.

Saeed et al. in [71] presented a framework for face detection in grayscale images. That framework relied on using the two dimensional central geometrical moments (2D-CGMs), up to the third-order, combined with the vertical and horizontal gradients of face components. The candidate regions were not selected but an exhaustive search over the space and scale is carried out by a multistage classifier which discards the background regions to focus on the possible face candidates. They also proposed a new method for the fast calculation of moments, and its complexity is invariant to the scale of the scanning window. The multistage classifier is a system of cascaded classifiers, where each classifier is a trained multi-layer perceptron neural network (MLPNN). The MIT CBCL database [82] was used for training and benchmarking. The highest accomplished detection rate was 99.07% at a 2% false positive rate.

In [85] Lin introduced an approach for the detection of faces in color images that is composed of two stages. The first stage involves converting the input RGB image into a binary image based on color segmentation using the relative ratio between the R, G, and B components. This is to eliminate the effect of lighting conditions. In the second stage, 4-connected components are found and labeled. The center of each block and the 3 other blocks are scanned to find the combination that forms an isosceles triangle. These components are potential face regions. The detected regions are resized into 60x60 pixels before being tested using a multilayer feed-forward neural network. The AR face database [13] was used for testing and benchmarking, and the approach achieved a 97.08% detection rate. No results were given for color images with complex backgrounds or skin-like backgrounds.

Stan et al. [88] presented a system which learned to detect multi-view faces within grayscale images using the Float Boost algorithm. The system uses coarse-to-fine architecture called detector-pyramid. Its significance was that it was the first real-time multi-view face detector running on 200ms per image whose resolution is 320x240 pixels on a Pentium-III CPU of 700 MHz. For frontal faces, the system accomplished a detection rate of 90.2% at a 6.44% false positive rate. The MIT and

CMU frontal data sets were used for testing. No results were mentioned on the multi-view face detection.

Chengjun Liu in [89] presented a novel method based on Bayesian Discriminating Features (BDF) for detection of frontal faces in grayscale images. Feature extraction from the input image involves deriving the BDF vector, 1-D Haar wavelet, and amplitude projection. Face and non-face classes were modelled as multivariate and normally distributed. Finally, Baye's classifier applied the estimated conditional PDF's to detect multiple frontal faces in the input image. Training used the FERET database [11] while testing was done using a subset of frontal faces (80 images containing only 227 frontal faces) that are collected from both the MIT [12] and the CMU datasets [3, 4]. The proposed method achieved a 98.5% detection rate and only one false positive case.

Sami et al. [90] ran an observation window at all possible positions, scales and orientations of an image to detect the faces. A non-linear SVM was used to determine if the window contains a face or not. The non-linear SVM works by comparing the input batch to a set of support vectors that are working as templates for face and non-face classes. The proposed system used the speeding-up method based on calculating a set of reduced vectors (RVs) sequentially from the support vectors. If the window contents are so unlikely to be a face, the remaining RVs are not calculated. This reduction in computation speeds up the system. Using the CMU dataset [3, 4] for testing, the system achieved a 80.7% detection rate at a false alarm rate of 0.001%.

A series of distinguished research was introduced by Viola and Jones [7, 91, 92]. They started in [7, 91] by introducing a machine learning based framework. They combined three contributions when they presented the integral image representation to compute the features very rapidly. They also used the Ada Boost algorithm to select a less number of critical visual features from a large set of initial features. Finally, they presented a combination of cascaded classifiers that are capable of eliminating the regions not of interest during the early stage to save the computational power for regions of actual interest. The framework ran at real time at 15 frames/second accomplishing a detection rate of 81.1% with a 0.02% false positive rate. It was tested using the combination of the MIT [12] and CMU [3, 4] datasets. In [92] they extended previous work to handle profile views and rotated faces. They built different detectors for each possible view of the face. A decision tree is then trained to determine the viewpoint class (right profile, rotated 30 degrees …) for the input image. The appropriate detector is then run instead of running the entire detector saving computational power and speed. The extended system accomplished a 70.4% detection rate at a 28.24% false positive rate. It proposed a 320×240 pixel image every 0.12 seconds running on 2.8 GHz Pentium-IV machine.

An impressive research was also presented by Yang et al. [93]. They presented two methods using multi-model density models for face detection in grayscale images. The first method uses a mixture of factor analyzers to perform clustering simultaneously with dimensionality reduction. The parameters of the models are estimated using the EM algorithm. The second method uses a Self Organizing Map for clustering, Fisher’s linear discriminant to find the optimal projection for pattern classification, and Gaussian model for the class-conditional density function of the projected samples for each class. The parameters of the Gaussian model are estimated

using the maximum likelihood and the decision rule is also based on maximum likelihood. Evaluation and testing was done using the MIT [12] and CMU [3, 4]. In the case of using CMU [3, 4] the first method achieved a 92.3% detection rate at a 16.97% false positive rate. But the second method achieved a 93.6% detection rate at a 15.32% false positive rate.

Recently, Byeong et al. [94] presented a face detection algorithm based on a first order reduced Coulomb energy (RCE) classifier. The algorithm locates frontal views of human faces at any degree of rotation and scale in a complex background. The face candidates and their orientations are first determined by computing the Hausdorff distance between simple face abstraction models and binary test windows in an image pyramid. Then, after normalizing the energy, each face candidate is verified by two subsequent classifiers: a binary image classifier and the first-order RCE classifier. While the binary image classifier is employed as a pre-classifier to discard non-faces with minimum computational complexity, the first-order RCE classifier is used as the main face classifier for final verification. Experimental results were obtained using set two of the CMU frontal face test set [3, 4] that has 223 frontal faces. The algorithm achieved a detection rate of 91% at no false alarm.

The most recent distinguished work was accomplished by Waring and Liu [95]. They introduced a face detection method using a spectral histogram and support vector machines (SVMs). Each image window is presented by its spectral histogram (it is a feature vector consisting of a histogram of filtered images). Using 4500 face and 8000 non-face samples, the SVM is trained to get a robust classifier. With an effective illumination-correction algorithm, the proposed system detects faces under different conditions. The system accomplished a detection rate of 96.67% with a 13.87% false positive rate when tested using the CMU database [3, 4]. But it achieved a detection rate of 95.6% with a 4.4% false positive rate.

Table 2.1 summarizes a comparison of the most significant research attempts in the field of face detection. These comparison results are collected from multiple comparisons in [1, 7, 95]. The comparison is reasonable because of using the same test datasets. Test set 1 is called CMU-125. Moreover, Test set 2 is called MIT-23. Finally, Test set 3 is called CMU-130.The specifications are described in detail in table 2.1 in chapter two of this thesis. Comparing these experimental results leads thinking to the fact that this is a relative comparison but no one can come out with an absolute decision that a specific technique is better than the others unless all the experiments have been carried out using the same dataset, the same evaluation procedure, and under the same conditions such as running on the same machines. This yields to the conclusion that the face detection research needs a standard evaluation procedure including a standard benchmarking image database. In addition, when comparing different face detection techniques, judging using the detection rate is not an enough indication because it is relative to the false positive rate. This is important criterion to consider when choosing a technique to be used for real life application, because the selection is dependent on the cost or risk function of false alarms within the target application.

TABLE 2.1
SIGNIFICANT FACE DETECTION METHODS

Method	Test Set 1 (483 FACES)		Test Set 2 (149 FACES)		Test Set 3 (507 FACES)	
	Detection rate (%)	False positive rate (%)	Detection rate (%)	False positive rate (%)	Detection rate (%)	False positive rate (%)
Integral Image and AdaBoost (Viola and Jones [7])	N/A	N/A	N/A	N/A	92.1	15.4
Integral Image and AdaBoost (Voting) (Viola and Jones [7])	N/A	N/A	N/A	N/A	93.1	15.4
Neural Network based (Rowley [3])	92.5	78.5	90.3	30.9	88.1	70
Naïve Bayes Classifier (Schneiderman and Kanade [97])	93.0	18.2	91.2	8.8	88.6	17.4
Distribution based (Sung and Poggio [12])	N/A	N/A	81.9	9.6	N/A	N/A
Mixture of factor analyzers (Ahuja et al. [98])	92.3	17	89.4	2.2	87.9	16.2
Fisher Linear Discriminant (Ahuja et al. [98])	93.6	15.3	91.5	0.7	89.2	14.6
SNoW with primitive features (Ahuja et al. [96])	94.2	17.4	93.6	2.2	89.7	16.6
SNoW with multi-scale features (Ahuja et al. [96])	94.8	16.1	94.1	2.2	90.3	15.4

Second, face detection systems that have been implemented in hardware for running in real-time. Some of these systems are summarized as follows: Irwin et al. presented a very impressive FPGA based system in [8]. They implemented the system presented in [4]. They focused on the design of special-purpose hardware for rotation-invariant face detection. The synthesized a design operating at 409.5 kHz providing 424 frame/second. It consumed 7 Watts of power. It also accomplished a detection rate of 75% comparative against the software implementation that gives a detection rate of 85% for the same test set.

Another implementation was described by Yu et al. [19]. They implemented the Ada Boost method that was introduced by Viola and Jones [7]. It operates at 91 MHz, and accomplishes 15 frames per second, where the frame size is 120×120 pixels. It was implemented using a Xilinx Virtex-2 FPGA using the Matlab System Generator design tool.

The next system was implemented by Hori et al. [78]. A 3D rational skin color model and a positive-negative lines-of-face template were presented to improve the signal to noise ratio (SNR) in face detection. A Steady State Genetic Algorithm (SSGA) was employed for lines-of-face detection from an entire image. The FPGA

based implementation was optimized for high speed and small hardware resources. It only consumed 40K gates of logic and 240K gates of memory. The system can operate at 30 frames per second with a resolution of 320×240 pixels. It accomplished a detection rate of 98% for a test set that contained 205 faces from 89 images. But the system's false positive rate was 18%.

An FPGA implementation of an integer MIPS processor was presented by Tirath et al. [82]. They used HANDEL-C to build the proposed system for face detection on a Celoxica RC2000 prototyping board. The system operates at 45 MHz. This is equivalent to approximately 13 frames per second when activating the noise reduction processing unit and 54 frames per second when it is not activated. The design consumes 115K NAND gates. No statistics were given on the accomplished detection rate.

Recently, another FPGA implementation was proposed in [79]. Melanie employed the reversible component analysis (RCT) for face detection. The FPGA implementation that was presented showed a negligible difference in performance between the hardware and software implementation. The Matlab environment was used for software implementation, where as a Xilinx Virtex-II FPGA on a Celoxica RC200 hardware prototyping board was used for hardware implementation. No information was given on the video rate but the system accomplished a detection rate of 74.1% at a false negative rate of 25.9%. The implementation required 255,416 NAND gates.

A recent system was presented by Duy et al in [80]. The authors present a new technique for face detection and lip feature extraction. The face detection classifier is based on naive Baye's classification. It classifies an edge-extracted representation of an image. Lip feature extraction uses the contrast around the lip contour to extract the height and width of the mouth. The proposed face detection system operates at 12.5 MHz giving 41.7 frames/second with a resolution of 320×240 pixels. It accomplished a detection rate of 86.6% at a zero false positive rate. The result was given when testing using the Yale dataset [81].

CHAPTER 3
Data Preparation for Benchmarking

3.1 Overview

The detection and recognition of human faces is one of the most important topics in computer vision and computational image analysis. The last decade has shown dramatic progress in this area, with emphasis on such applications as human-computer interaction (HCI), biometric analysis, content-based coding of images and videos, content-based image retrieval systems, robotics vision and surveillance systems.

The availability of large sets of images is essential in these fields for the development, testing, and analysis of new techniques. For human face recognition, several large image databases are commonly available (e.g., [11, 13, and 30]). The images in such databases typically show human subjects conveniently posed and properly illuminated, and with each face positioned in the centre of the image and appropriately scaled. On the other hand for face detection, relatively few image databases are available to the research community. This might seem surprising, because ultimately, the detection step must be performed prior to further processing that involves the face. Unfortunately, the lack of a standard data set represents a substantial burden for researchers. Without reference test cases, it is difficult to develop evaluation criteria for the benchmarking of new face detection algorithms and implementations.

As shown earlier in chapter two, face detection algorithms and techniques are classified into several categories. No matter what class of algorithm is under consideration, a benchmarking data set has to satisfy many criteria. For example, the number of images must be adequate for both training and testing, and the images should present many different poses, backgrounds, lighting conditions, facial expressions, and variations of skin color. Databases with a wide variety of these attributes will help in assessing particular algorithms, and in making fair comparisons among different techniques.

3.2 Previous benchmarking datasets

This section assesses existing databases, and describes some advantages as well as disadvantages of each. Several grayscale face image databases have been compiled. Among them are the well-known CMU test set (Rowley, et al. [3, 4]) and CMU profile test set (Schneiderman and Kanade [5]), which have been used for benchmarking purposes (e.g., [6, 7, 8]).

The CMU test set provides the researchers with 507 frontal faces, in which some are rotated and thus not upright. It also supplies the user with a solid ground for each face location. All the images are in CompuServe GIF format, and are grayscale. CMU profile test set that was compiled later included non-frontal profile faces and rotated faces. However, there was no available statistical information on how many faces belong to each category. Figure 3.1 shows some examples taken from the CMU datasets. Another database, the FERET database was presented in both grayscale and

color. The grayscale FERET database is available now as bonus data for the users but the color FERET database is the main scheme. It presents a single face per image with multiple poses (frontal, left profile, and right profile) for the same person. It is more adequate for testing face recognition techniques than face detection techniques.

The UMIST face poses database [86] and the AT&T database [87] are other examples of grayscale databases. The AT&T face database contains a set of face images taken between April 1992 and April 1994 at the AT&T lab, in Cambridge University. There are ten different images of each of 40 distinct persons. For some persons, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The files are in PGM format. The size of each image is 92x112 pixels, with 256 grey levels per pixel. The images are organised in 40 directories (one for each person), which have names of the form `sX`, where `X` indicates the person number (between 1 and 40). In each of these directories, there are ten different images of that person, which have names of the form `Y.pgm`, where `Y` is the image number for that person (between 1 and 10).

There have been very few attempts to construct databases of color face images. One of those is the Kodak face image database (Loui, et al. [9]), with 80 images and with 90% of the faces shown in a frontal pose. Moreover, some details about the AR database (Martinez and Benavente [13]), CVL Face database [30], and UOPB Face Database [31] are also worth mentioning.

Figure 3.1: Sample images from the MIT [12] and CMU test sets [5].

The Kodak database is divided into two datasets, a data set used for face detection and another data set used for face recognition. Each picture is available in two formats: the Kodak FlashPix format and the Tagged Image File Format (TIFF) in order to label the picture with input meta-data. The main disadvantage here is that

90% of the available faces are frontal. Such frontal cases are not enough for the accurate evaluation of a new face detection technique.

The AR database was compiled in the Computer Vision Center (CVC) at Purdue University. It is named after Aleix Martinez and Robert Benaventi. It is also known as the Purdue database. It displays a single person per image. Different facial expressions, illumination conditions, and occlusions are depicted to collect 26 different snapshots per person. This database is more appropriate to study the effects of facial expression and illumination conditions on the processing of human faces. A sample from the AR database is shown in Figure 3.3.

The CVL database was compiled in the Faculty of Computer and Information Science (FRI) in the University of Ljubljana, Slovenia. It includes 798 images that are saved in JPEG format with a resolution of 640x480 pixels. Each picture has a single face and each person has 7 pictures with different poses and the facial expressions (frontal, left profile, right profile, smiling, etc.). This database is adequate for the evaluation of face localization algorithms. A sample from the CVL database is shown in Figure 3.2.

The University of Oulu's Physics-Based Face Database is another face database [31]. It has 125 various faces each having 16 different shots. The shots differ according to various camera calibrations and illumination conditions. The additional 16 shots are added if a structural component like a glass exists in the picture. All faces are in a frontal position and they are captured under particular illuminations (horizon, incandescent, fluorescent and daylight). It also offers 3 spectral reflections of skin per person that are measured from the cheeks to the forehead. Each image has a resolution of 428x569 pixels in 24-bit RGB, stored in BMP-format.

The UCD color database is considered the closest attempt to build a standard database for the testing of face detection techniques. The UCD color database was compiled by Sharma and Reilly [10]. It contains a total of 299 human faces where there is a wide variety of poses, orientation, backgrounds, facial expressions, structural components, image quality, occlusion, age, race, size and gender. It is divided into two parts. The first part consists of 94 color images. The other part of this database contains the manually segmented skin regions in the images in the first collection. A sample from this database is shown in Figure 3.4.

Figure 3.2: Sample images from the CVL Color Face Image Database [30].

Figure 3.3: Sample images from the AR Color Face Image Database [13].

Figure 3.4: Sample images from Set1 and Set 2 of the UCD Color Face Image Database [10].

Table 3.1 summarizes all the databases that were mentioned in the literature. The list includes only the databases that are still available for distribution or available online. The list includes databases for training, testing, and evaluation purposes.

TABLE 3.1

FACE DETECTION DATABASES

Each is described in terms of the number of images and file formats, as well as attributes of the image content such as pose and number of faces present.

Database	Description
FERET Database (Philips, et al.[11])	14,051 eight-bit grayscale images of human heads with views ranging from frontal to left and right profiles.
MIT Database (Sung and Poggio [12])	Set 1 includes 301 grayscale frontal and almost frontal face mugshots for 71 persons. Set 2 (also called MIT-23) has 23 images containing 149 faces.
MIT CBCL Face Database [82]	It is divided into a training set that includes 2429 faces and 4548 non-face objects and a testing set that includes 472 faces and 23573 non-faces. All pictures are 19x19 pixels in PGM format. it is available for online download.
CMU Frontal Face Test Set (Rowley, et al. [3, 4])	It is divided into set one that includes 507 frontal faces from 130 grayscale images (also called CMU-130), this set includes 23 images of the second dataset of MIT database. When subtracting 5 images including hand drawn faces, the total faces are 483 faces. This is called (CMU-125). In addition, set two consists of 50 grayscale images containing 223 faces, of which 210 are at angles of more than 10 degrees.
CMU Profile Face Test set (Schneiderman and Kanade [5])	208 greyscale images including 347 profile views plus some frontal faces.
Yale Face Database (A. Georghiades [81])	It contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: centre-light, w/glasses, happy, left-light, with/without glasses, normal, right-light, sad, sleepy, surprised, and wink.
Kodak Database (Loui, et al.[9])	80 color images in TIFF format. Approximately 10% of the faces are not frontal. The face size ranges from 13×13 pixels to 300×300 pixels.
AR Database (Martinez and Benavente [13])	This set includes 4,000 color images with faces of 126 individuals (70 men and 56 women). It uses a single person per image on a simplified uniform background. Format: TIFF, 768×576, 24 bits/pixel.
CVL Face Database[30]	798 color images with faces of 114 persons, 7 images of each person. All are in JPEG format, and size of 640*480 pixels.
UOPB Face Database [31]	125 various faces each having 16 different shots under different camera calibrations and illumination conditions. A single image is 428x569 pixels in 24-bit RGB, stored in BMP-format.
UCD Database (Sharma and Reilly [10])	94 color images in GIF format. Contains 299 faces with 182 frontal poses, 91 intermediate poses, and 26 profiles. Wide variety of face size, 3D pose, orientation, and occlusion.
UMIST Face Poses Database [86]	It consists of 564 images of 20 people. Each covering a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. Each subject exists in their own directory labeled (1a, 1b... 1t) and images are numbered consecutively as they were taken. The files are all in PGM format, approximately 220 x 220 pixels in 256 shades of grey. Pre-cropped version is also available.
AT&T Face Database [87]	It contains 400 images of 40 persons. There are ten different images for each person. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). The size of files is 92x112 pixels in PGM format, The size of each image is, with 256 grey levels per pixel.

3.3 The need for a standard benchmarking database

Most of the existing databases that are listed in Table 3.1 have some drawbacks when used for the purpose of face detection. For example, a traditional approach to displaying human faces is to show one person per image against a simple, uniform background. Most of the FERET, Kodak, AR and CVL images follow this convention. Although this approach is helpful for face recognition, many face detection applications require an ability to accommodate more complex scenarios. Another aspect of many face databases is that they contain grayscale images only. This is quite limiting for applications that require skin detection, which often relies heavily on color-based analysis (e.g., [14, 15, 16, and 17]).

To the best of our knowledge, the UCD color database [10] is the only publicly available database that satisfies the requirements of color imagery plus a variety of face poses, and background scenery. Unfortunately, the UCD color database also has few shortcomings. For example, it has a relatively small number of faces, only 299 faces. It doesn't provide the researcher with any statistical information about the variables that define the level of difficulty in detecting the faces within each image. For example there are no records for the number of occluded or rotated faces per each image. The difficulty level can be determined based on the earlier knowledge of the pose, orientation, background, facial expressions, structural components, image quality, occlusion, age, race, size and gender. The UCD suffers from the lack of an appropriate filing system. There is no explicit database file that includes all the information that is required by researchers for the easy implementation of the evaluating and benchmarking system.

It is a clear now to the reader how the face detection field is in real need for a standardization process. This standardization process must include the compilation of a special color image database for evaluation and benchmarking of face detection techniques and algorithms. Such a database can be also used for testing human skin segmentation techniques. The next section proposes a new color image database that considers the previously mentioned drawbacks and takes into account most of the suggested recommendations. It is under the name VT-AAST image database.

3.4 Compilation of a new benchmarking database: VT-AAST Color Image Database

This section describes the process of assembling and compiling the VT-AAST database in detail. The VT-AAST image database is a new color face database for the benchmarking of automatic face detection algorithms and human skin segmentation techniques**.** It is divided into four parts. Part one is a set of 286 color photographs that include a total of 1027 faces in the original format produced by digital cameras, offering a wide range of difference in orientation, pose, environment, illumination, facial expression and race. Part two contains the same set in a different file format. The third part is a corresponding set of image files that contain color human skin regions resulting from a manual segmentation procedure. The fourth part of the database has those same regions converted into grayscale. The database is available on-line for noncommercial use.

The images were captured using several consumer-grade digital cameras from different vendors. Image-array sizes ranged from 3 to 5.2 megapixels. Part of the reasoning behind using these "point-and-shoot" cameras instead of professional-grade imagers was to mimic expected applications of face-detection systems. We assume that most of these applications rely on relatively inexpensive cameras, such as those mounted on laptop computers, or those typically used for surveillance in public places.

Our cameras provided images in JPEG format. (Most of today's consumer-grade cameras do not provide images in uncompressed form.) These original images are given in part one of the database. These images were further compressed to the Graphics Interchange Format (GIF), with 300×225 pixels per image to form part two of the database. Therefore, each photograph is provided in both JPEG and GIF formats. In addition, a manual segmentation procedure was used to identify skin regions for each case. These segmented images are available in two forms: with color information retained for pixels with the skin regions, and with only grayscale information retained for the skin regions. All of the segmented images are stored as GIF files of size 300×225. These segmented cases represent the third and fourth parts of the VT-AAST database, respectively.

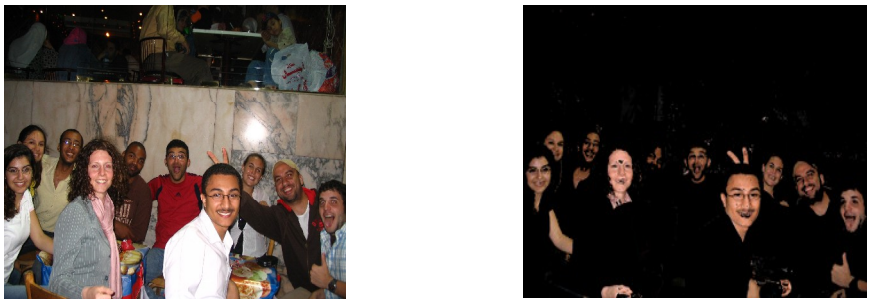
A sample from the database is shown in Figure 3.5.The choice of image size matches those commonly used for video analysis (e.g., [8, 23, and 29]). Figure 3.5 illustrates the results of the manual segmentation process.

The database currently contains 286×4 images, where the factor of 4 represents the number of different image formats available for each original photograph. These photographs were obtained in both indoor and outdoor environments, against a variety of backgrounds. Several samples taken from the database are shown in Figure 3.6. The goal was to provide a wide variety of the following:

- 3-dimensional pose (frontal, intermediate, profile, and "over-profile").
- 2-dimensional orientation (upright and rotated).
- Facial sizes in the images.
- Facial expressions.
- Luminance.
- Occlusion.
- Structural components (hair, beards, mustaches, glasses).
- Gender.
- Human race (White, Black, Asian, etc., as specified by the U.S Census Bureau [18]).
- Scene and background (outdoor and indoor, simple and cluttered).

Table 3.2 provides a breakdown of these categories, as presented in the database. For the 3D poses, we follow the definitions given by Reilly [10]. Essentially, for a frontal view the subject faces the camera, and for a profile view a plane parallel to the image plane divides the face equally. The intermediate pose is sometimes called a three-quarter view, and indicates that the head has been turned

slightly away from the camera. Finally, the over-profile view refers to the case that the subject is looking away from the camera, so that the ear and cheek are visible but the tip of the nose is not.



(a) (b)

Figure 3.5: Sample images from the VT-AAST database. (a) Color image, available in both JPEG and GIF forms. (b) Corresponding segmented image in which the skin regions have been retained, but all other pixels have been converted to a dark background value. The segmented image is also available with the skin regions represented using grayscale pixel values. These segmentation steps were performed manually for all images in the database.

TABLE 3.2

STATISTICAL DATA FOR THE VT-AAST IMAGE DATABASE

The 286 photographs contain 1027 instances of human faces in a variety of backgrounds and environments.

Total Number of Images = 286																					
Total Number of Faces = 1027																					
3D Pose				Orientation		Gender		Race						Structural Components		Occlusion					
Frontal	Intermediate	Profile	Over profile	Upright	Rotated	Male	Female	White	Black	American Indian	Asian	Hawaiian	Other	Sunglasses or Glasses	Mustache or Beard						
515	390	91	29	885	142	532	495	828	102	1	0	0	93	334	142	106					

Structural components typically refer to obstructions that partially cover the face. Typical examples include facial hair and eyeglasses. Racial information is categorized in the database according to the latest definitions given by U.S Census Bureau [18].

In order to store image details, we used both Microsoft Excel [26] and the Microsoft Access database engine [27]. Our database consists of a single table whose public key is the image ID; each image entry has specific fields that correspond to one of the attributes described above. Figure 3.7 presents a sample record, with values shown for the different fields.

Figure 3.6: More example images from the VT-AAST database. (a-c) Many are group pictures, providing several faces for detection, in a variety of indoor and outdoor environments. (d) Unusual orientations are present in some of the images, along with a mixture of structural components such as sunglasses and facial hair. (e) Different 3-D orientations of the individuals' heads present are shown. (f) Three-dimensional perspective may cause faces to appear at dramatically different sizes in the images.

The Excel spreadsheet consists of three columns. The first column contains a unique identifier for each set of 4 images, corresponding to 1 photograph. The second column states the total number of faces in each image. In some cases, a few face regions of an image are not included in this count. This occurred when a face was overlooked during the manual segmentation step due to color and brightness patterns that caused it to blend into the background. To indicate such cases, a "+" symbol appears in the 3rd column of the spreadsheet. The manual extraction was done using the color range selection tool in Adobe Photoshop CS2 [22]. The main reason for using the Microsoft tools was to facilitate integration with the Matlab computing environment [28]. Matlab's Visual Query Builder provides a capability for database querying, and Matlab's Excel Link tool allows the incorporation of Matlab commands within Excel spreadsheets.

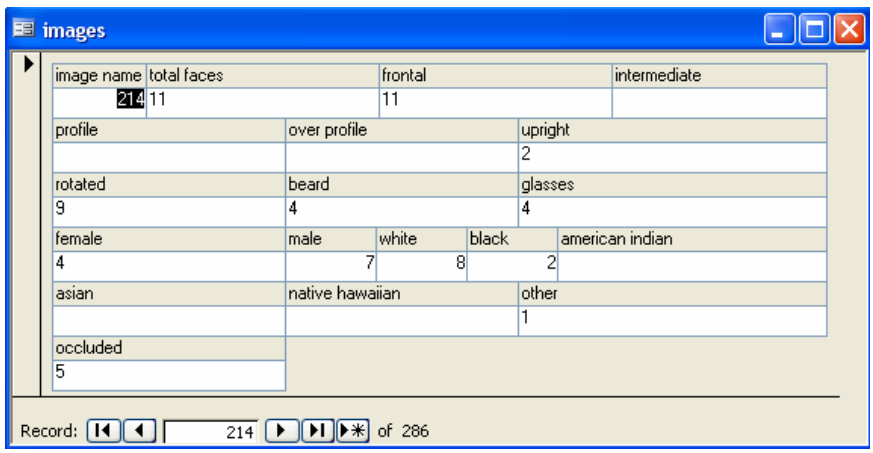


Figure 3.7: Sample record from the VT-AAST database. This type of tabulated information is provided for every image in the database. Blank fields represent values of zero.

The number of faces in the images is relatively large, and therefore facilitates the identification of separate training and testing sets for benchmarking of face detection approaches. Examples of approaches include support vector machines (e.g., [20, 21]), artificial neural networks (e.g., [3, 4, and 16]), AdaBoost (e.g., [19]) and FloatBoost ([24, 25]). A wide range of skin colors appear in the images, and this presents a challenge to skin-detection algorithms, as well as an opportunity for further research.

The VT-AAST database can serve as a standard for comparison for future face- and skin-detection research. Access to the images is provided through the Internet. They are currently available at http://filebox.vt.edu/users/yarab/VT-AAST Database, although a password is needed to access the site. Access will be provided to users who sign an agreement for non-commercial use of the images.

This section has introduced a new image database that fills an important need for researchers who are interested in human face and skin detection. Created as a joint effort by Virginia Tech and the Arab Academy for Science, Technology, and Marine Transport, the VT-AAST color image database is large enough to be used for both training and testing. The images are sufficiently varied in content to provide a wide range of poses, skin colors, illumination conditions, and background complexities. This is distinguished from previous databases by the extensive tabulation of image content (number of faces for each image, etc.), and by providing manually segmented "ground truth" images of skin regions for each case.

The VT-AAST database is the main and only source of samples for training and testing of the new techniques or approaches that are going to be presented among the rest of this thesis.

CHAPTER 4
Theoretical Background

4.1 Overview

This thesis introduces face detection techniques that are significantly appropriate for hardware implementation. These techniques therefore have the potential for use in commercial embedded systems. The face detection process will be partitioned into the following sub phases:

Pre-processing stage is the stage where the input image is processed to extract the regions of interest for later processing using the main feature extraction algorithm. The goal of this stage is to avoid processing the areas not of interest to the main algorithm.

Feature extraction stage is the stage responsible for finding a set of common attributes among all the patterns such that the range of values of these attributes must be close for patterns of the same class, while it has to vary between the patterns of the different classes. Therefore the classification boundary can be determined. Traditionally, this is done by employing a priori knowledge of the problem domain to specify some facial features like the eyes, the nose, and the mouth. But the traditional feature extraction methods have limitations. Therefore, another category has shown up which is called image based analysis methods such as the Fourier transform, and other mathematical image analysis approaches.

Dimensionality reduction stage. One problem of using image based methods for feature extraction is the potential for a high dimensional feature space. For practical systems, dimensionality reduction is needed to reduce the required computational power and memory consumption.

Classification and decision making stage refers to the process of grouping different items into separate sets based on specific criteria. These criteria are some attributes related to the processed items. The process requires enough number of training samples either labeled or non-labeled.

4.2 Pre-processing of the input image

Traditional methods of face detection that have been developed generally for offline processing used to scan the input image on a multi-resolution base. The input image is processed using exhaustive multi-resolution approach. The output is typically multi-resolution image pyramid that consists of thousands of blocks to be processed later by the main detection algorithms. Chapter 5 of this thesis presents a non-traditional preprocessing technique that is based on the following fundamentals.

4.2.1 Skin Segmentation

Human skin has unique color properties. The idea of using skin color in the pre-processing stage of face detection has been suggested because it is invariant against scaling, rotations, translation, and skewing. The basic two challenges are deciding which color space is to be used, and how to model the skin color distribution. Skin detection methods are categorized into two categories [32]. Both

pixel based skin detection methods and region based skin detection methods are discussed below.

4.2.1.1 Pixel-based skin detection

Pixel-based skin detection depends on classifying each pixel independently from its neighbors. A skin color detector can be defined in terms of the relation between the chrominance and intensity components, the hue, saturation components, or the red, green and blue components. This varies according to the used color space. For example, Peer et al. [33] used an RGB space where the skin pixel was identified using the following decision rule:

$$\begin{aligned} &\textit{If } (r > 95 \,\&\, g > 40 \,\&\, b > 20 \\ &\quad \&\ (\,(\max(r,g,b) - \min(r,g,b)) > 15\,) \\ &\quad \&\ (\,|\, r - g \,|\, > 15\,) \,\&\, (r > g) \,\&\, (r > b)\,) \\ &\textit{Then}\,\{\, X(r,g,b) \text{ is a skin pixel } \} \end{aligned} \tag{4.1}$$

Pixel based skin detection may use either a parametric or non-parametric skin distribution models. In case of a parametric modeling, a predefined statistical model is selected to model the skin distribution. The most common models are listed in [32]. They are single Gaussian, mixture of Gaussians, multiple Gaussian clusters and an elliptic boundary model. Most are based on the chrominance component, discarding the luminance. In the case of non-parametric modeling, a histogram analysis of the training data follows to come up with a skin probability function that describes the skin distribution. Later, a probability value is given to each pixel. Several implementations are done for the non-parametric methods using normalized lookup tables (LUT), Baye's classifier, and self organizing maps (SOM) [32].

Parametric modeling is more sensitive to the choice of color space than the non-parametric modeling because of the effect of the distribution shape on the curve fitness criteria. On the other hand, non-parametric models are not only independent of the shape of skin distribution, but also they are faster in training and testing. Unfortunately they require more memory and there is no way to generalize the training data.

4.2.1.2 Region-based skin detection

Region based skin detection considers the spatial arrangement of skin pixels. Several methods were mentioned in the literature. For example, Kruppa et al. presented in [34] a parametric model combining skin color and shape. It depends on the existence of a model-generated skin pixel distribution and the distribution of skin color that is found in the input image. The presented model describes the spatial arrangement of skin pixels as elliptical regions. The basic idea is to minimize mutual information between the two distributions.

Yang and Ahuja [35] proposed a region-based skin detector that was used for face detection directly. They proposed a parametric skin color model that is based on chrominance information using multivariate statistical analysis. Multiple skin regions

were extracted firstly using multi-scale image segmentation by a special mapping function. These different regions were then merged to get the elliptical shape that is described by the model's parameters. The proposed model was tested using CIE-LUV color space.

Jedynak et al. [36] demonstrated three models. The first model is pixel-based, the second one is a hidden Markov model that considers the spatial arrangement of skin pixels using the 4 neighbors system. Third, a color gradient based model was proposed, taking into account the relationship between the skin and non-skin neighboring pixels. Analytical expressions were given for the coefficients of the collaborated maximum entropy model.

4.2.2 Morphological Operations

The word morphology means the scientific study of forms, shapes and structure. It differs from branch to another based on the object under study. If it is an animal or plant, then the branch is called biology. If the object is a word then the branch is called Linguistics. Moreover, if the object is found in a digital image, then we call this branch morphological digital image processing.

The morphological operations are a set of digital image processes that are based on the concept of mathematical morphology. Mathematical morphology is a mathematical model that depends on Minkowski addition and set theory. Morphological operations were firstly used on binary images and were then extended for usage on grayscale images. For binary images, the sets are modeled in the integer two dimensional (2D) space Z^2, where each pixel is a set of two coordinates $P(x,y)$. As for gray images, the sets are modeled in a three dimensional space where the first two coordinates represent its position on the image plane and the third is its integer discretized grayscale value.

Morphological operations are very useful in describing a region shape in an image. It is widely used for preprocessing operations like filtering, thinning, thickening, and pruning. The dilation and erosion are the main operations that are used in combination with the set theory to introduce a long list of morphological operations for binary images. Examples of morphological operations include opening, closing, the hit-or-miss transform, boundary extraction, region filling, extraction of connected components, convex hull, thinning, thickening, skeletonization, shrinking, and pruning. In addition, they are employed within grayscale image applications such as morphological smoothing, morphological Gradient, top-hat transformation, textural segmentation, and Granulometry. The basic morphological operations are defined as follows:

Dilation is also known as Minkowski addition It is the operation of adding every element in set A to every element in set B as shown in (4.2)

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \tag{4.2}$$

It can also be defined in terms of reflection and translation as shown in (4.3)

$$A \oplus B = \{z \mid [(\hat{B}) \cap A] \subseteq A\} \tag{4.3}$$

Set *B* is called the morphological structuring element. The above equation describes the dilation as being composed of two stages, determining the reflection of the structuring element B about its origin. This reflection is shifted by z. Finally, the result of dilation is the set that includes all the displacements between *A* and *B*, where an overlap takes place within at least one element

.

Erosion is the process of finding all the pixels that are determined by translating the structuring element *B* by z, such that the shape *A* contains all these pixels z. The morphological mathematical definition is given by

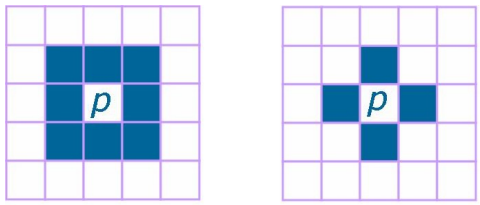
$$A \ominus B = \{Z \mid (B)_z \subseteq A\} \tag{4.4}$$

Note that dilation and erosion are twofold; they can be expressed in terms of complementation and reflection as shown in (4.5)

$$(A \ominus B)^c = A^c \oplus B^c \tag{4.5}$$

4.2.3 Region Analysis

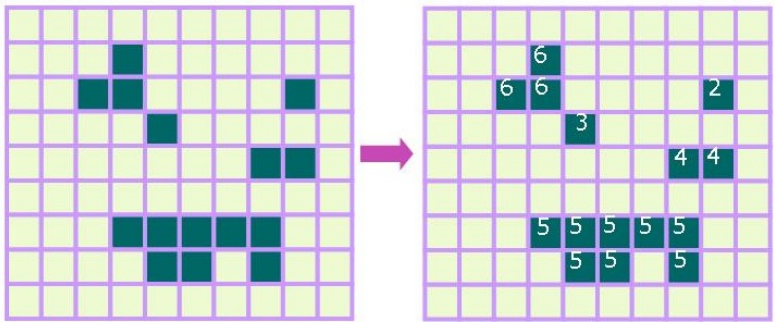
A region is defined as a connected component of an image. A connected component of an image is a set of pixels, such that every pixel of the set is connected to every other pixel of the set according to a predefined connectivity scheme. But this is not an absolute definition because it is based on the definition of connectivity. In other words, it is based on the used connectivity scheme. There are several connectivity schemes between pixels in an image. Figure 4.1 shows an example of two connectivity schemes, the 4-neighbor scheme and the 8-neighbor scheme.



(a) (b)

Figure 4.1: Example of a connectivity scheme. (a) Eight neighbours. (b) Four neighbours.

Based on the previous definition of a region, the region labelling process is defined as the act of assigning a unique label, most probably a unique integer value, to a set of pixels that belong to the same region. An example of an input image contains multiple regions and the corresponding output in the case of region labelling using 4-neighbor connectivity is shown in Figure 4.2.



(a) (b)

Figure 4.2: Example of a connectivity scheme. (a) An input image. (b) The result of region labelling.

4.3 Feature Extraction

The feature extraction process is accomplished taking into account that two conditions have to be met. The first condition is the need to select a set of features that can clearly distinguish the objects belonging to different classes. This is available when the decision boundaries don't cause an overlap between the different classes. But this is a very ideal theoretical assumption that is very hard to be satisfied in practice. Therefore a compromise is usually needed in pattern recognition problems to get the highest possible detection rate. The second condition is that the set of selected features should have feasible requirements in terms of computational power and memory consumption. This why some feature extraction techniques are only appropriate for offline processing on mini-computers while others are well suited for real-time running on embedded systems. Below is a short introduction on some of the useful feature extraction methods.

4.3.1 Discrete Cosine Transform

The Discrete Cosine Transform is widely used for different applications. The most popular use of the DCT is in data compression, and it forms the basis for the well-known JPEG image compression format [39, 42]. The DCT coefficients can also be used as a type of signature that is useful for recognition tasks, such as facial expression recognition [40, 41]. Conceptually, each DCT coefficient can be viewed as a representation of a different feature dimension.

4.3.1.1 Definition

The Discrete Cosine Transform can be defined as the Discrete Fourier Transform of roughly twice the length, using real numbers data with even symmetry. The two dimensional Discrete Cosine Transform (2D-DCT) is an extension of the single dimensional discrete cosine transform (DCT) that is given by

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \tag{4.6}$$

The 2D-DCT can be calculated by applying the 1D-DCT in consecutive order. A row based DCT is applied on the two dimensional signal such as a digital image, followed

by a column based DCT. The same operation is directly expressed using double summation by (4.7),

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2x+1)v}{2N}\right] \qquad (4.7)$$

where $\alpha(u)$ and $\alpha(v)$ are given by

$$\alpha(u) = \begin{cases} \sqrt{1/N} & \text{for } u = 0 \\ \sqrt{2/N} & \text{for } u \neq 0 \end{cases} \qquad (4.8)$$

Figure 4.3 demonstrates the two dimensional DCT basis functions for $N = 8$. In the top left corner presented in gray, the DC component can be noticed; it is a constant function. The frequency increases in the both directions, the vertical and horizontal. The white color represents the positive amplitudes, while the black color represents the negative amplitudes.

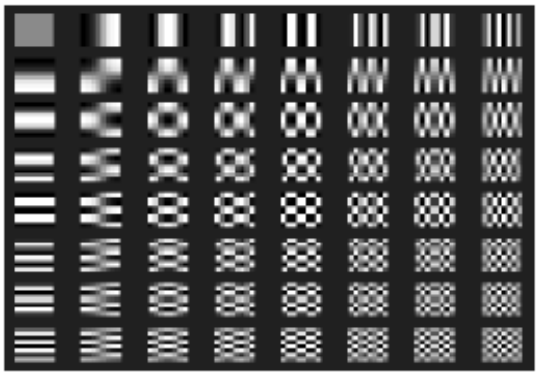


Figure 4.3: Two dimensional DCT bases functions at ($N = 8$).

4.3.1.2 Properties of the DCT

The two dimensional discrete cosine transform (2D-DCT) has several properties. These properties are described here.

Decorrelation, The main goal of image transformation is getting rid of redundant information. The two dimensional discrete cosine transform (2D-DCT) coefficients have uncorrelated characteristics.

Energy Compaction, The DCT has an excellent capability of energy compaction for correlated images. The more uncorrelated the input image becomes, the higher frequency components it contains. This leads to distributing the image energy on larger range in the frequency space. In the case of a correlated image, the energy is condensed on small range in a region of low frequency. A result of this energy compaction property is that a small number of 2D-DCT coefficients, is enough to reconstruct the original image using the 2D-IDCT. This is the main advantage of the 2D-DCT in the field of image compression.

Separability, This property is inherited from the ability to calculate the 2D-DCT output as a process pf two stages, a row based 1D-DCT followed by a column based 1D-DCT. It is expressed mathematically by

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\left[\cos\left[\frac{\pi(2x+1)u}{2N}\right]\left(\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2y+1)v}{2N}\right]\right)\right] \quad (4.9)$$

Orthogonality, The DCT basis functions are orthogonal. This means none of these functions can be represented as a combination of other basis functions. They are independent of each other. The advantage of this property is the reduction in the pre-computation complexity.

Symmetry, This is also inherited form the ability to calculate the 2D-DCT output as a combination of row based followed by column based 1D-DCT. The two stages are identical. This property, in addition to the separability property, reveals that the transformation matrix can be calculated offline for any input image, simplifying the computational complexity of the problem.

4.3.2 Discrete Wavelet Transform

In this section a short introduction on the discrete wavelet transformation (DWT) is presented using the simple Haar wavelet as an example. Mathematically, the Discrete Wavelet Transform is a wavelet transform, such that the wavelets are discretely sampled. The wavelet transform is based on the idea of representing a function (signal) in terms of scaled and shifted waveforms of finite length. These waveforms are considered as basis functions also known as mother wavelets. The represented function must be square-integrable and the wavelets have to be either a complete orthonormal set of basis functions or an over-complete set of frame functions. A wavelet transform differs from a Fourier transform whose basis functions are sinusoids because the wavelets have varying frequencies and confined durations. This makes them able to localize not only what frequencies are played but also the time where different frequencies are played. On the other hand, the Fourier transform and similar transforms like the cosine or sine transforms can only provide the frequency information.

4.3.2.1 One dimensional discrete wavelet transform

The Haar wavelet was presented by Alfréd Haar; it is inherited from the Haar transform [45]. For a discrete signal f of length N, the output of the Haar transform is two signals each one having half the length of the original signal. These are an approximate signal a of length $N/2$ and signal of details d whose length is also N/2.

For example, if the signal f is defined as follows:

$$f = (2,6,4,12,20,10) \quad (4.10)$$

Then the approximate calculated signal will be:

$$a = (2\sqrt{2}, 8\sqrt{2}, 15\sqrt{2}) \tag{4.11}$$

And the calculated signal of details will be:

$$d = (-2\sqrt{2}, -4\sqrt{2}, 5\sqrt{2}) \tag{4.12}$$

The details signal is calculated from the scalar product of the signal f and the Haar wavelets set W which is defined by:

$$\mathbf{w} = \begin{bmatrix} w_1^1 \\ w_2^1 \\ . \\ . \\ . \\ w_{N/2}^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & . & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & . & . & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \tag{4.13}$$

In addition, the approximate signal is given by the scalar product of the signal f and the scaling signal v that is defined by:

$$\mathbf{v} = \begin{bmatrix} v_1^1 \\ v_2^1 \\ . \\ . \\ . \\ v_{N/2}^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & . & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & . & . & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{4.14}$$

This wavelet based analysis can be repeated for multiple levels by applying the same scalar product on the approximated signal given by the level before. For level (l), the approximated signal is given by:

$$a^l = a^{l-1}.\mathbf{v} \tag{4.15}$$

Here the signal of details of the same level is given by:

$$d^l = d^{l-1}.\mathbf{w} \tag{4.16}$$

The most important properties of the Haar discrete wavelet transform as well as other discrete wavelet transforms are the conservation and compaction of energy. Note that the energy of any signal is the sum of the squares of this signal. Regarding the energy conservation property, the total energy after the transformation is the sum of the energy maintained by the approximate signal and the saved energy in the signal of details. This total is equal to the total energy of the original signal. This can be formulated as follows:

$$Energy(f) = Energy(a^1) + Energy(d^1) \tag{4.17}$$

As for energy compaction property, the total energy that is maintained by the approximate signal saves a large percentage of the energy of the original signal. Therefore a large percentage of the original signal can be restored. Notice that both the Haar transform and the inverse Haar transform can be defined in a mapping form that is formulated as:

$$f \underset{H^{-1}}{\overset{H}{\rightleftarrows}} (a^1, d^1) \tag{4.18}$$

4.3.2.2 Two dimensional discrete wavelet transform

The two dimensional Haar transform can be defined easily in terms of the one dimensional Haar. 2D-DWT consists of two phases. The first phase is applying the single dimensional DWT on the two dimensional input signal in row/column based order, where as, the second phase is applying the single dimensional DWT on the output of the first phase in column/row based order. The 2D-DWT can be defined based on a cascading filtering scheme. In such a case it will be defined as a two stage filtering process, the first stage consisting of two filters - a low pass filter and a high pass filter. Then each filter is followed by another two filters, a low pass and a high pass filter. Figure 4.4 demonstrates the 2D-DWT filtering scheme, while Figure 4.5 shows an example of an input image and its corresponding Haar wavelet decomposition up to level 2. It also shows the corresponding outputs from the filters.

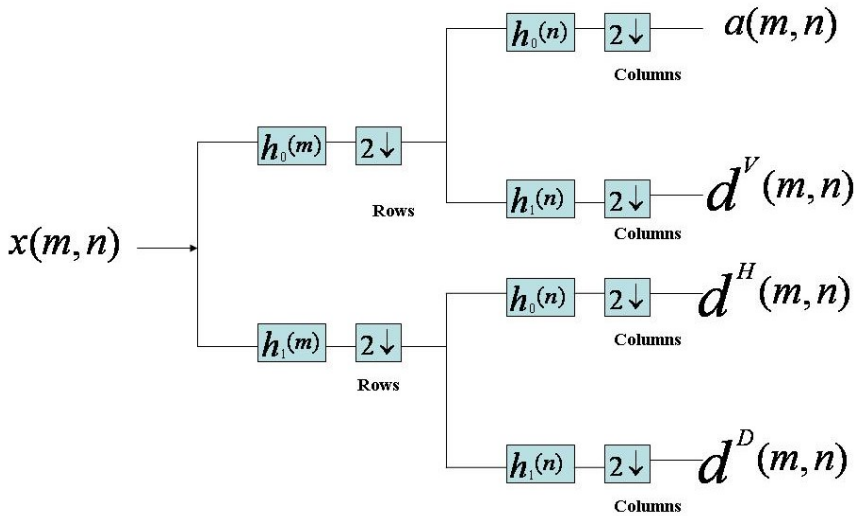
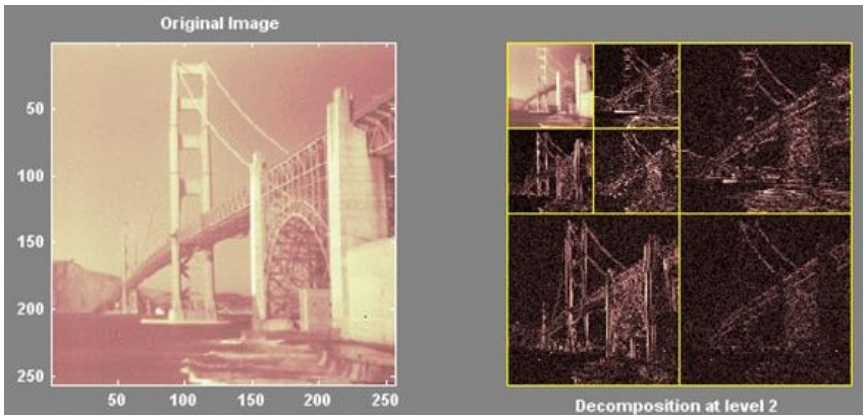
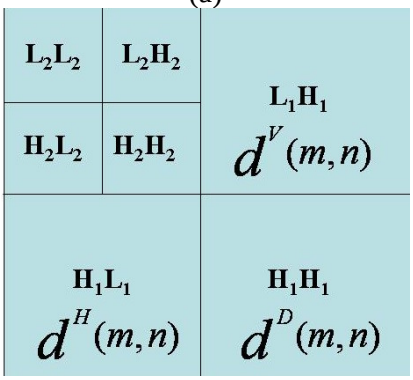


Figure 4.4: Two dimensional DWT based on cascading filtering scheme.



(a)



(b)

Figure 4.5: An example of an input image and its corresponding output of the Haar wavelet transform. (a) The input image and its Haar decomposition up to level 2. (b) The corresponding output in terms of four band filter bank for image decomposition.

The general properties of the wavelet basis functions are summarized below. The different families of wavelets satisfy different combinations of the following properties [46, 47, 48]. A good understanding of these properties helps the researchers to select the wavelet family that is most appropriate for the target application.

Orthogonality directly links the L^2 norm of a function to the norm of its wavelet coefficients by (4.19).The advantage of orthogonal wavelets is that the fast wavelet transform is a unitary transformation. Consequently, its condition number is equal to one, which is the optimal case. In the bi-orthogonal case, these two quantities are equivalent.

$$\| f \| = \sqrt{\sum_{j,1} \gamma_{j,1}^2} \tag{4.19}$$

Compact support If the scaling function and wavelet are compactly supported, the filters *h* and *g* are finite impulse response filters, so that the summations in the fast wavelet transform are finite. This is obviously of use in implementations. If they are not compactly supported, a fast decay is desirable so that the filters can be approximated reasonably by finite impulse response filters.

Rational Coefficients. For hardware implementations, it is of use if the filter coefficients h_k and g_k are rationales or, even better, dyadic rationales. Multiplication

by a power of two corresponds to shifting bits, which is a very fast operation in terms of hardware implementation.

Symmetry. If the scaling function and wavelet are symmetrical, then the filters have a generalized linear phase. The absence of this property can lead to phase distortion. This is important in image analysis applications.

The *smoothness* of the wavelet plays an important role in image compression applications. Compression is usually achieved by assigning small coefficients in equation (4.19) to zero. If the original function represents an image and the wavelet is not smooth, the error can easily be detected visually. Furthermore, a higher degree of smoothness corresponds to better frequency localization of the filters.

The *number of vanishing moments* is a very important property for singularity detection and characterization of smoothness spaces. Also, it is used to determine the convergence rate of wavelet approximations of smooth functions. In addition, the number of vanishing moments of the dual wavelets is connected to the smoothness of the wavelet (and vice versa).

Table 4.1 summarizes the main properties of the wavelets families that are going to be used in this thesis, namely; Haar, Daubechies-*p*, Coiflets-*p*, and symlets-*p*.

TABLE 4.1

MAIN PROPERTIES OF DIFFERENT WAVELET FAMILIES

Each is described in terms of orthogonality, compact support, symmetry, smoothness, and number of vanishing moments

The name of the Family of Wavelets	Relevant properties
Haar	Orthogonal
	Compact support
	The scaling function is symmetric
	The wavelet function is anti-symmetric
	It has only one vanishing moment (a minimum)
Daubechies-*p*	Orthogonal
	Compact support
	There is no symmetry for $p > 1$
	It has P vanishing moments
	Filter length is $2p$
Coiflets-*p*	Orthogonal
	Compact support
	Almost symmetric
	The wavelet function has $2p$ vanishing points and the scaling function has $(2p-1)$ vanishing moments
	Filter length is $6p$
Symlets- *p*	Orthogonal
	Compact support
	Almost symmetric
	It has P vanishing moments
	Filter length is $2p$

4.3.2.3 Applications

The Discrete Wavelet Transform (DWT) is becoming one of the most powerful tools for showing an input image in a time-frequency representation. This is the case when time localization of the different frequency components is essential. The DWT arranges a set of high pass and low pass filters in multi-rate organization [43]. DWT treats an input image as 2D signal so that it first traverses the image row-wise before traversing it column-wise. The discrete wavelet transform is widely used for video and image compression, particularly MPEG-4 and JPEG-2000. The DWT has several wavelet families such as: the original Haar Mother Wavelet, the Daubechies family of wavelets [44], the Coiflet family of wavelets, and Symmlet wavelets. The extracted DWT coefficients based on any one of previously mentioned wavelets can be used as a kind of signature that is useful for recognition tasks. Conceptually, each DWT coefficient can be viewed as a representation of a different feature dimension.

4.3.3 Geometrical Moments

In physics, the moment of a force is a quantity that represents the magnitude of force applied to a rotational system at a distance from the axis of rotation. The moment arm distance is the key to the operation of lever, pulley, gear and other machines capable of generating a mechanical advantage. Mathematically, the concept of moment matured over time from the physical concept of moment. The *n*th moment of a real valued function $f(x)$ of a real variable x about a value c is given by:

$$\mu_n' = \int_{-\infty}^{\infty} (x-c)^n \, f(x) \, dx \tag{4.20}$$

Moments for random variables are defined more generally than moments for real values. The moments about zero are referred to as the moments of a function. The function is usually a probability density function. The nth moment (about zero) of a probability density function $f(X)$ is the expected value of X^n.

4.3.3.1 Significance of the moments

The moments of $f(X)$ about its mean μ are called central moments; these describe the shape of the function independently of any translation effect. Any set of moments that satisfies this translation independence feature is called translation invariant moments or translation invariants.

The first moment about zero, if it exists, is the expectation of X, i.e. the mean of the probability distribution of X, designated by μ. In higher orders, the central moments are more interesting than the moments about zero. The *n*th central moment of the probability distribution of a random variable X is given by:

$$\mu_n = \mathrm{E}((\mathrm{X} - \mu)^n) \tag{4.21}$$

The first central moment is thus 0; the second central moment is the variance, the square root of which is the standard deviation, σ. The normalized nth central moment is the nth central moment divided by σ^n; the nth moment of $t = (x - \mu)/\sigma$. These normalized central moments are dimensionless quantities, which represent the distribution independent of any linear change of scale. Any set of moments that satisfies this scaling dependence criterion is called scaling invariant moments or scaling invariants.

The third central moment is a measure of the lopsidedness of the distribution; any symmetric distribution will have a third central moment, if defined, of zero. The normalized third central moment is called the skewness, often γ ("normalized" in this case means divided by the cube of the standard deviation σ^3, thus making the skewness dimensionless). A distribution that is skewed to the left (the tail of the distribution is heavier on the left) will have a negative skewness. A distribution that is skewed to the right (the tail of the distribution is heavier on the right), will have a positive skewness.

4.3.3.2 Invariant moments for image processing

A large number of moments that are invariant under several geometrical transformations such as: rotation, scaling, translation, and translation were proposed in the literature. Hu was the first who presented invariant moments [49]. He presented a set of invariant moments under rotation, scaling, and translation that are applicable on grayscale images. In [50], Abo-Zaid et al. focused more on the problem to come up with invariant moments under photometric scale changes. On the other hand, Abu-Mostafa and Psaltis [51] focused on rotation invariance and noise robustness. Recently, more interest was directed to use the color information without taking into account the shape information [52, and 53]. However, recently in [54], a complete set of independent moments able to describe a shape are presented by Flusser. They are applicable on grayscale images. He showed that there is a small set of moments that form a basis for all the rotation invariant moments. Bidoggia and Gentili [55] followed Flusser by presenting a set of moments that are a basis for the moment invariants for rotation, scaling, translation, reflection, and photometric scale changes. Those moments also have a good discrimination power for contrast changes and compression of the image.

An important issue that is worth to be addressed is the effect of digitization on the performance of moments since the moments are originally defined in a continuous domain. Gouda and Abbott addressed later the effect of spatial quantization on several moment invariants [56]. They presented an analysis of quantization-induced error on (two- dimensional) Hu moment invariants and affine moment invariants (AMIs), as well as on invariants derived from (one-dimensional) contour moments. Error bounds were given in several cases.

An efficient shape descriptor plays an important role in the feature extraction process. There are two types of shape descriptors: contour-based shape descriptors and region-based shape descriptors [57]. Hu moment invariants are some of the most famous, widely used and easy to implement contour-based shape descriptors. Hu [49] defined seven moments computed by normalizing central moments up to the third

order; those are invariant to object scale, position, and orientation. The definitions of Hu moment invariants in terms of normalized central moments are given below:

$$
\begin{aligned}
H1 &= \eta_{20} + \eta_{02} \\
H2 &= (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \\
H3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
H4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
H5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
H6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\
&+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})] \\
H7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&- (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned}
\tag{4.22}
$$

Where η_{pq} is given by normalizing the central moments μ_{pq} as follows, given that $\gamma = [(p+q)/2]+1$ and $p, q = 0, 1, 2, 3, \ldots$.

$$
\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{4.23}
$$

Figure 4.6 shows examples of the set of geometrical transformations those Hu moments are invariant under their effects.

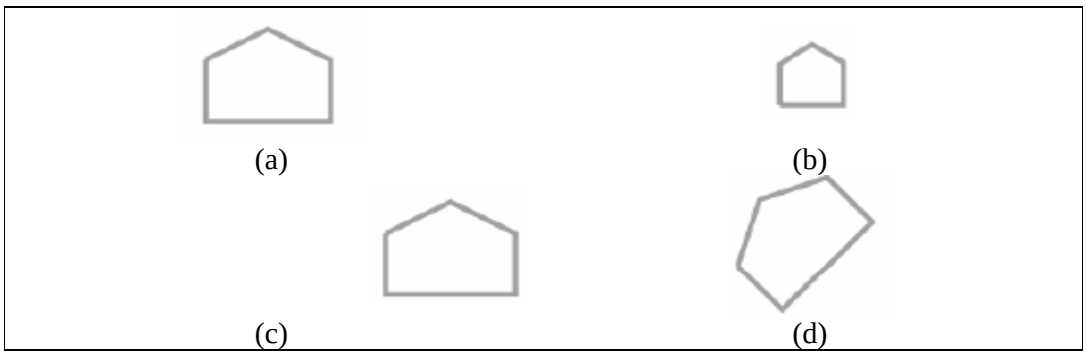


Figure 4.6: An example of geometrical transformations. (a) The original shape. (b) The same shape after a scaling effect. (c) The same shape after a translation effect. (d) The same shape under rotation.

4.3.4 Edge Detection

4.3.4.1 Overview

It is the process of indicating the pixels in a grayscale image where the intensity value changes suddenly. This is one of the traditional techniques of feature extraction. It plays an important role because sudden changes in intensity value are indications for possible important events such as: discontinuity in the depth, discontinuity in the surface orientation, changes in the texture of the visible objects, or variation in the luminance of the surrounded environment.

Edge detection methods can be classified into two categories, search-based edge detection and zero-crossing edge detection [59, and 60]. Search-based methods depend on the localizing of maxima and minima in the first derivative of the input image. Alternatively, zero-crossing methods depend on searching for zero crossings in the second derivative of the input image. Either the zero crossings of Laplacian or non-linear differential expressions are usually used.

The edge detection techniques can also classified based on what features are to be detected into:

Detecting of step edges. The first order partial derivative is computed by approximating the gradient of the image intensity function at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. Estimation of the gradient vector is based on the use of (3 X 3) masks. Sobel's masks and Prewitt's masks are the most well known operators. Prewitts's mask is given by:

$$\Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \Delta_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.24}$$

Sobel's mask is defined by:

$$\Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \Delta_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.25}$$

Both Prewitt's and Sobel's edge detectors are derived based on the assumption that white noise is additive and that image surfaces are linear. This is why they are sensitive to noise and they both don't perform well in case of noisy input images.

Detecting of lines. The lines are correspondent to local maxima of the intensity level of the input image; they may represent roads, rivers, or runways in satellite images. Detecting of lines is highly dependent on a preceding stage where a thinning algorithm must be applied earlier on the input image. It has non-accurate results especially with complex scenes like satellite images.

Detecting of junctions. A junction is defined as the intersection of two linear step edges.

4.3.4.2 Canny edge detection algorithm

The algorithm was presented by John F. Canny [58]. Canny aimed to come up with an optimal edge detection technique. In his paper, he followed a list of criteria to improve current methods of edge detection:

Good detection. The first and most obvious criterion is a low error rate. It is important that the edges occurring in images should not be missed and that there be NO responses to non-edges.

Good localization. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels are as found by the detector and the actual edge is at a minimum.

Minimal response is to have only one response to a single edge. This was implemented because the first 2 were not sufficient enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria; the Canny edge detector first smoothes the image to eliminate the noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the second threshold. The steps of the Canny edge detection algorithms are demonstrated in more details below:

Noise reduction. Using a Gaussian mask, the input image is convolved, so that the output looks like a blurred copy of the original in order to reduce the effect of the single noisy pixel. The larger the width of the Gaussian mask, the lower the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width increases.

Finding the intensity gradient of the image. An edge in an image may point in a variety of directions, so the Canny algorithm uses four masks to detect horizontal, vertical and diagonal edges. The result of convolving the original image with each of these masks is stored. For each pixel, we then mark the largest result at that pixel, and the direction of mask which produced that edge. From the original image, we created a map of intensity gradients at each point in the image, and direction of the intensity gradient points. The magnitude of the gradient is then approximated by:

$$|G| = |Gx| + |Gy| \tag{4.26}$$

While the edge direction is given by:

$$\theta = \tan^{-1}(\frac{Gy}{Gx}) \quad (4.27)$$

Based on the value of θ, the detected direction will be linked to a direction from the four possible directions that can be tracked in an image (North, South, East, and West).

Tracing edges through the image. After the edge directions are known, non-maximum suppression is then applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered an edge. This will give a thin line in the output image. The higher intensity gradients are more likely to be edges. There is not an exact value at which a given intensity gradient switches from not being an edge to being an edge. Therefore Canny uses thresholding along with hysteresis. Thresholding with hysteresis requires two thresholds - high and low. Making the assumption that those important edges should be in continuous lines through the image, allows us to follow a faint section of a given line, but avoiding the identification of a few noisy pixels that do not constitute a line. Therefore we begin by applying a high threshold to mark out the edges we can be fairly sure are genuine. Starting from these, using the directional information derived earlier, edges can be traced through the image. While tracing a line, we apply the lower threshold, allowing us to trace faint sections of lines as long as we find a starting point. Once this process is complete, we have a binary image where each pixel is marked as either an edge pixel or a non-edge pixel. Figure 4.6 shows an example of an image and the result of applying the Canny algorithm on it.

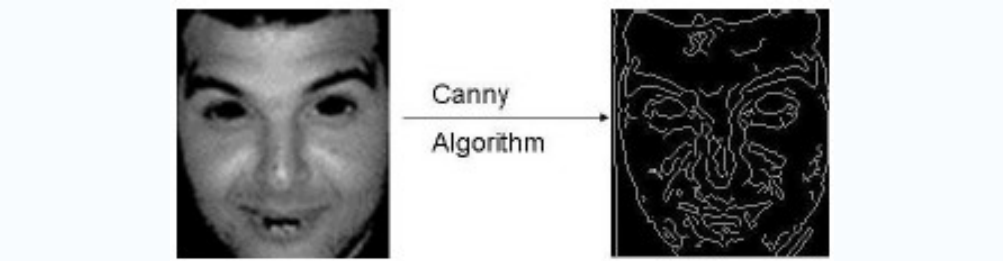


Figure 4.7: An example of Canny edge detection. For an input of a gray scale image and the corresponding response of the Canny edge detection algorithm.

4.3.5 Dimensionality Reduction

4.3.5.1 Overview

Statistically, the dimensionality reduction is the process of reducing the number of random variables in the problem space under specific considerations. Physically, it is the phenomenon, where a physical system is defined on a space whose dimension is *d* but it behaves like a system whose dimension is *p*, given that *p* is less than *d*. In practice for pattern recognition field, this can be interpreted as having a pool of samples (observations). For each sample we have a set of features *D*. The target is to select a subset *A* of features from the large set *D*, such that the significant attributes of each sample are captured so that each one still can be

distinguished from the others. Notice that the optimum is to make the size of *A* minimum in order to save more computational power and memory requirements [61].

The dimensionality reduction may enhance the system performance because it keeps only the non-correlated features while removes any redundant information. However, the system performance may decrease because it removes also a significant part of the available information per processed sample. The issue of performance here is application dependent.

4.3.5.2 Principal Component Analysis

The principal component analysis (PCA) is the best linear technique for dimensionality reduction [61]. It has been used in different applications; this is why it is also known as the Karhunen-Loeve test, the Hotelling transform, the empirical orthogonal function (EOF), as well as singular value decomposition (SVD). PCA is based on calculating the covariance matrix of the variables.

The target of PCA is to represent the given data using a less number of components (PCs) using a linear transformation. The original set of variables *D* whose size is l, is transformed linearly into another set of orthogonal components whose maximum size is l too. Mathematically, the problem is modeled as follows: We have an l dimensional feature vector $\mathbf{V} = (v_1, ..., v_l)^T$ with mean $\mathrm{E}(\mathbf{V}) = \boldsymbol{\mu} = (\mu_1, ..., \mu_l)$ and the covariance matrix $\boldsymbol{\Sigma}_{lxl} = \mathrm{E}\{(\mathbf{v} - \boldsymbol{\mu})(\mathbf{v} - \boldsymbol{\mu})^T\}$. We aim to find $\mathbf{S} = (s_1, ..., s_k)$, such that the $\max(k) = l$. $\mathbf{S}$ is given by the linear transformation:

$$\mathbf{S} = \mathbf{W}\ \mathbf{V} \tag{4.28}$$

The first component $s_1 = \mathbf{V}^T W_1$ is composed of the linear combination with the highest variance. The second component $s_2 = \mathbf{V}^T w_2$ is the linear combination whose variance is the second largest variance and it is also orthogonal to the first component. The process is repeated to get the first k components, such that the l-dimensional transformation vector $W_i = (w_{i,1}, ..., w_{i,l})$ satisfies the largest variance criterion given by (4.29) and $i \in \{1, 2, ..., k\}$

$$W_i = \mathrm{argmax}_{\|W=i\|}\ \mathrm{var}\{\mathbf{V}^T \mathbf{W}\} \tag{4.29}$$

Because it is highly probable that not all the features presented in $\mathbf{V}$ have the same scale, there is a need to normalize or standardize the data. This need can be satisfied easily by using the correlation matrix instead of using the covariance matrix where the correlation matrix is calculated by dividing each element $\sigma_{i,j}$ in $\boldsymbol{\Sigma}_{lxl}$ by $\sqrt{\sigma_{i,i}\sigma_{j,j}}$. Then the eigenvalues and the eigenvectors for the correlation matrix $\mathbf{R}_{lxl}$ are calculated. It can be rewritten using the spectral decomposition theorem in the form $\mathbf{R}_{lxl} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_l)$ the diagonal matrix that contains the eigenvalues in ascending order and $\mathbf{U}$ is the orthogonal matrix that

contains the corresponding eigenvectors. $\mathbf{S}$ can then be determined now using (4.30). Notice that the first k eigenvectors have the smallest mean square deviation from $\mathbf{V}$ across all the subspaces of dimension k.

$$\mathbf{S} = \mathbf{U}^T\mathbf{V} \tag{4.30}$$

Now the first k components can be used as a new set of features for further processing saving computational power and memory. In this thesis, another method is adopted [62]. This PCA test is used to select the most important features from the original feature vector $\mathbf{V}$. First, all the l possible principal components will be obtained. Next, pick the eigenvector corresponding to the smallest eigenvalue which is the least important principal components and discard the feature that has the largest absolute variable in the vector. Then, pick the eigenvector that is corresponding to the second smallest eigenvalue, and get rid of the feature that has the largest absolute value in the vector, given that it was not thrown away before. Finally, after $(l-k)$ iterations, keep the remaining k features to be used as the fine selected set of features.

4.4 Learning Based Classification

4.4.1 Introduction

Classification is the process of organizing a large set of samples into n smaller subsets based on specified criteria. Automating the classification process is one of the greatest applications of the related research to artificial intelligence.

The ability to learn is a basic characteristic of intelligent creatures. Artificial neural network is one of these learning based methods that is based on the learning theory. The idea of artificial network was originally inspired of the neural networks in the human brain. Although an accurate definition doesn't exist, a learning process in an artificial neural network (ANN) context can be viewed as the problem of updating the network architecture and connection weights so that a network can perform well a specific task. The ANN must learn the connection weights from an available set of training patterns. The performance improves overtime by iteratively updating the weights in the network. Instead of following a set of rules like case based reasoning systems, the ANNs learn what is underlying the rules using the learning algorithm. It is the procedure that describes the way in which a set of learning rules are used to adjust the connection weights. There are three main learning paradigms: supervised, unsupervised and reinforcement learning.

Supervised learning. The network is provided with a correct answer for every pattern (it is the label of the true class that this pattern truly belongs to it). Weights are determined to allow the network to generate the answers as close as possible to the known correct answers.

Unsupervised learning. In contrast with the previous type, there is no associated label with the presented pattern in the training dataset. The underlying structure of the data is explored, or the correlation between the patterns in order to classify the patterns into subsets (categories) based on these correlations.

Reinforcement learning. It is a combination of the supervised and unsupervised learning. The network is provided with only an analysis on the correctness of network outputs, not the correct answers themselves. In other words, the system is not supplied with external indications for what the correct responses must be or whether the generated outputs are right or wrong.

There are three fundamentals that must be taken into account when talking about the learning theory for artificial neural networks. The first issue is the capacity that concerns the number of patterns that can be stored and what functions and decision boundaries the network can perform on. The next one is the sample complexity concerning how to determine the lower bound and the upper bound of the number of training patterns that is needed to guarantee a valid generalization. Finally, the computational complexity of the system must be considered such as the time required for the learning algorithm to estimate the solution from the training dataset.

Competitive learning is one of those unsupervised learning schemes. In this scheme only one output unit can be activated at any particular time. The output units compete with each other so that the winner is activated. The winner at any particular time is the unit that has the highest value of the predefined winning criterion at that time. The winning unit is also known as the best matching unit (BMU). The winning unit adapts its weight vector to be better matched with the input vector. This is done repeatedly during the learning phase. A Self organizing map (SOM) that is also known as Kohonen's self organizing feature map or Kohonen's neural network is one of the most famous competitive learning techniques.

4.4.2 Self Organizing Map Neural Network

The self organizing maps (SOM) were proposed by Kohonen in [65, 66] and described thoroughly in [64, 63]. They are deemed to be highly effective as a sophisticated visualization tool for visualizing high dimensional complex data with inherent relationships between the various features comprising the data. The output of the SOM emphasizes the salient features of the data and subsequently leads to the automatic formation of the clusters of similar data items [67]. The SOM has been employed in a wide range of applications, ranging from financial data analysis [68], via medical data analysis, to time series prediction, industrial control, and many more [116].

This particular clustering characteristic of SOM makes them qualified candidate to work as a clustering classifier after setting a specified scheme to define the decision boundaries. A learnt SOM can be used as a powerful clustering tool since the output is organized, such that similar data items are grouped together. This is an additional application in addition to its original task to work as a data visualization tool that maps a high dimensional input data x, $x \in \Re^n$ onto a, usually two dimensional, output space while preserving the topological relationships between the input data item as genuine as possible. But for practical implementation usage of an SOM for clustering, it is still required for marking the output layer of the SOM to visualize the distinguished clusters of the similar patterns. This is an ad hoc operation but with the proposed visualization method called U-matrix [69]. This process is managed automatically.

The U-matrix method uses the distances between the units of the SOM as boundary definition criteria. These distances can then be displayed as heights giving a U-matrix landscape. The U-matrix is interpreted as follows: the altitudes or the high places on the U-matrix will encode data that are dissimilar while the data falling in the same valley will represent data that are similar, so that data that are in the same valley can be grouped together to form a separate cluster.

The SOM contains *N* nodes ordered in a two-dimensional lattice structure. Both rectangular and hexagonal lattices are tested. In these cases, each node has 4 or 6 neighboring nodes, respectively. Typically, the SOM has a life cycle of three phases: the initialization phase, the learning phase, and the testing phase. The codebook vector for each node can be initialized randomly or linearly. During training, the learning function is responsible for updating the codebook vectors of the neurons that are located in predefined neighborhoods of the winning neuron. The most widely used training function is a parameterized Gaussian function,

$$h_{ci}(t) = \lambda(t).e^{(-\frac{\|x_c - x_i\|}{2\sigma^2(t)})} \tag{4.31}$$

where $\lambda(t)$ is the learning rate and $\sigma(t)$ represents the radius of the neighborhood set of nodes. The winning metric is usually based on calculating a Manhattan or Euclidean distance function between the input vector and each entry in the node's codebook:

$$\| x - v_c \| = \min_{1 \le i \le N} \arg\{\| x - v_i \|\} \tag{4.32}$$

Both $\lambda(t)$ and $\sigma(t)$ decrease as the training progresses. Finally, testing is performed by comparing the error computed for each input feature vector against a specified threshold. For demonstration purposes, Figure 4.7 shows an example of the U-matrix of an SOM, the U-matrix is color based on the topological distance criteria in order to illustrate the decision boundaries among different clusters. In case of supervised learning, the labels of classes can be used instead of colors.

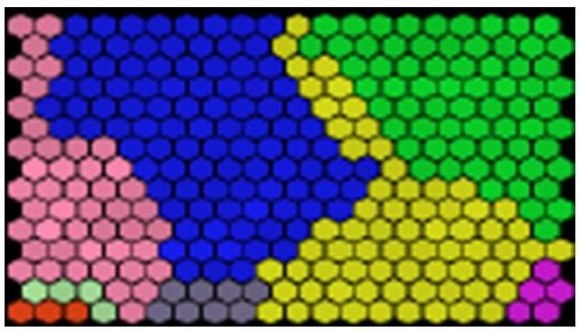


Figure 4.8: An example of the output layer of an SOM. The different clusters are visualized using the U-matrix method. Each color corresponds to a different class of data.

Any SOM based classifier can be modified easily to be trained using a supervised session instead of using its default unsupervised learning algorithm. This is done by considering the class label of each pattern as an additional dimension within

the problem space given this new variable a weight that is equal to the weights of the rest of variables. The supervised classification process using SOM has three phases:

Supervised training. During the training phase, labeled hybrid feature vectors are presented to the SOM one at a time. For each node, the number of "wins" is recorded along with the label of the input sample for each win. The codebooks for the nodes are updated as described above. By the end of this stage, each node belonging to the SOM has two recorded values: the total number of winning times for facial input samples, and the total number of winning times for non-facial input samples.

Voting. A simple voting approach is used to assign a label to each node, such that the label represents the class that is associated with the largest number of winning times. In case of a tie, the processed node receives a blank label. This is also true for nodes having zero winning times.

Testing phase. During the testing phase, each input vector is compared with all nodes of the SOM, and the best match is found based on minimum distance, as given in (2). Euclidean distance was used in our tests. The final output of the system was the label associated with the winning node.

CHAPTER 5
Experimental Results

5.1 Introduction

This chapter presents the core of the research. First, it introduces a novel scheme that combines basic image processing methods (including skin segmentation, morphological operations, and region connectivity analysis) for color images to specify the regions that are most likely to contain human faces. Then, the chapter describes a list of experiments covering the approaches, observations, and interpretations. Each experiment has two phases. The first phase presents a novel approach for the construction of a feature vector for each candidate skin region. Several feature extraction methods have been investigated. Next, the second phase of each experiment involves using the feature vectors that were built during the earlier phase to train a learning based classifier. The classifier in use is a Self Organizing Map (SOM). Different results were collected for each experiment based on changing different parameters of feature extraction methods and of the SOM based classifier. The presented work forms a fully integrated system for face detection. A block diagram of the proposed system, starting after skin segmentation, is shown in figure 5.1

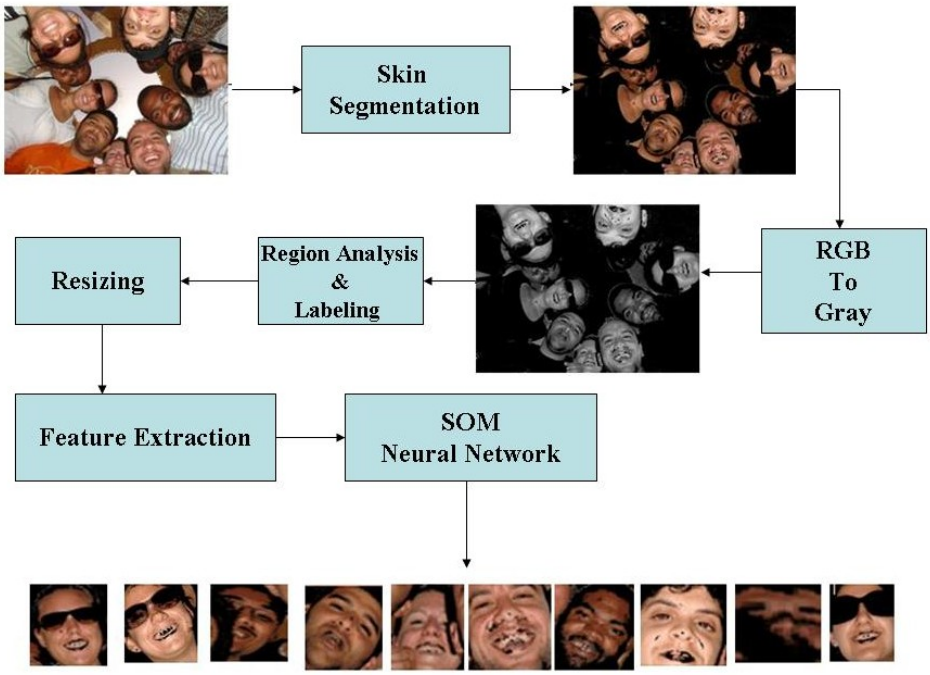


Figure 5.1: Block diagram of the face-detection system. The input is a segmented "skin map." After a region analysis step, features are extracted for skin regions. Faces, if present, are then detected using a self-organizing map.

All of the experiments were performed after passing the data through the same preprocessing stage, except for minor differences that were necessary according to the experimental context. The evaluation of detection methods and comparative results

are based mainly on calculating the detection rate, false positive ratio, and false negative ratio. They are given by:

$$Detection\ Rate\ (\%) = \frac{N_S}{N_F} \times\ 100 \tag{5.1}$$

$$False\ Positive\ Rate\ (\%) = \frac{N_{FP}}{N_{NF}} \times\ 100 \tag{5.2}$$

$$False\ Negative\ Rate\ (\%) = \frac{N_{FN}}{N_F} \times\ 100 \tag{5.3}$$

In these equations, N_F is the total number of facial regions, N_S is the number of correctly detected faces, N_{NF} is the total number of skin regions that don't contain faces (non-facial regions), N_{FP} is the number of non-facial regions that are detected incorrectly as faces, and N_{FN} is the number of facial regions that are detected incorrectly as non-facial.

Training, testing, and analysis have been accomplished using the VT-AAST color face database. It is divided into two subsets, for separate training and testing of the system. During SOM training, 129 images were used, containing 439 facial skin regions and 770 non-facial skin regions. The testing phase used 157 images with 588 facial and 1369 non-facial regions. None of the images that were used for training have been included into the testing set.

The face detection methods presented in this thesis were developed, trained, and tested using MATLAB™ 7.1. The computer used was a Windows XP machine with a single 2.00 GHz Centrino processor and 512 MB of RAM. The SOM work has been implemented and developed using the SOM Toolbox [99].Training was performed using the batch algorithm after modification to make the process supervised as described in subsection 4.4.2 of this thesis. Unless otherwise stated, this was the method used in all of the following experiments.

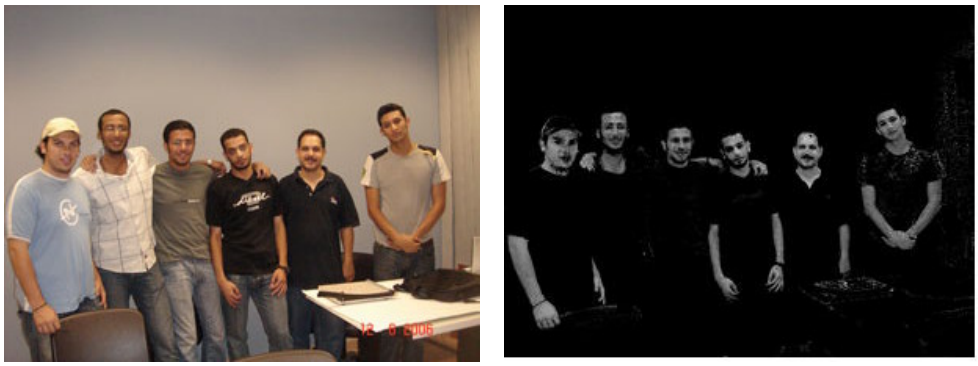
5.2 Preprocessing Stage

The input of this stage is a color image, encoded in JPEG form

5.2.1 Skin segmentation process

The input color RGB image (I) of size ($M \times N \times 3$) is processed pixel by pixel using (4.1) to distinguish the pixels that are likely to be human skin from those that are not. The output is a binary image (I_b) of size ($M \times N$). Pixels whose values are 1 represent the locations of skin and values of 0 stand for background pixels. Simultaneously, (I) is converted into grayscale giving (I_g) which is of the same size of the input image. The binary image is (I_b) masked with the grayscale version (I_g) to obtain a new image showing the intensity values of the skin pixels only. This

grayscale skin image will be referred to as a "skin map" throughout the rest of this thesis. Figure 5.2 shows an example of the input and output of the skin segmentation process.



(a) (b)

Figure 5.2: Example of skin segmentation for face detection. (a) Color input image. (b) Segmented image, known as a skin map with skin pixels represented using intensity values.

5.2.2 Morphological Opening process

The input to the region analysis stage is a skin map in which pixel values of the detected skin regions are represented using intensity values. For simplicity of implementation, the value zero (black) is assumed to represent background pixels. In this stage a morphological opening operation is applied to the skin map. The opening step modifies the boundary of a region slightly, typically reducing its curvature. More importantly for this application, morphological opening has the effect of subdividing large regions that are connected by a narrow connecting path, sometimes called a "neck." This has the effect of separating candidate face regions from other body parts, thereby improving the chances of correct detection.

5.2.3 Labeling of Connected Components

The input of this process is a modified skin map that resulted from applying the morphological opening process in the previous stage. Labeling of connected components is a procedure that assigns a unique label to each region in the image. In effect, each skin region receives a unique index that aids in further analysis. Region labeling in this system is done using eight-neighbor connectivity. A common alternative would be to use four-neighbor connectivity instead, but this was not tested in our system. The use of the eight-neighbor connectivity scheme has the advantage of generating smaller number of candidate facial regions.

The final output of this stage is a list of labeled connected regions. Each is a skin-like region that is a candidate human face, and grayscale information is provided for each. In the case of using 2D-DCT, 2D-DWT, or edge detection for feature extraction, these regions are resized to fit into a standard pre-chosen block whose size is 32 × 32 pixels. Resizing is done using the nearest neighbor interpolation algorithm. In the case of using geometrical moments, the regions are left with no resizing. The resizing operation is the last of the preprocessing operations performed on the input image. So now the input image is ready for further processing to extract the needed features for the construction of the feature vector for each candidate region.

5.3 Experimental Results

5.3.1 Feature Extraction using 2D-DCT

In this section, the 2D-DCT coefficients are calculated for each skin block, resulting from the previous stage. This results in a matrix of 32 × 32 coefficients. A subset of these values is taken to construct the feature vector. Empirically, the upper left corner of the 2D-DCT matrix contains the most important values, because they correspond to low-frequency components within the processed image block. For the initial experiments presented below, we used a set of 16 × 16 coefficients.

This section presents the results of four experiments in which different system parameters were altered. In the first experiment, the effect of the size of the SOM was studied in an effort to find the optimum grid size. Table 5.1 shows the detection results for this experiment, including detection rates and false positive rates for different sizes of the SOM network. Each of these networks was trained using a sample array of size 1209 × 256 including the face and non-face training samples, and each was tested using a sample array of size 1957 × 256 including the face and non-face testing samples. The best detection rate obtained here was 77.7% for an SOM of 100 nodes arranged in a 10 × 10 hexagonal lattice. This SOM size was used for all subsequent experiments. Training was performed using the batch algorithm, and a total of 3000 epochs.

TABLE 5.1
THE EFFECT OF NETWORK SIZE ON THE SELF-ORGANIZING MAP (SOM)
The SOM size versus detection results for three different sizes.

	SOM Structure		
	Size = 50 (5 × 10)	Size = 100 (10 × 10)	Size = 150 (15 × 10)
Detection rate (%)	74.8	77.7	73.3
False positive rate (%)	6.0	5.1	9.5

The second experiment studied the effect of the DCT block size on the accuracy of detection, with each DCT coefficient used in the feature vector. Table 5.2 shows that slightly better performance was obtained for the case of 16 × 16 block sizes, and this block size was used in all subsequent experiments. This is also shown graphically in figure 5.3.

TABLE 5.2
DETECTION RATES VS. DCT BLOCK SIZE
Maximum detection rate is obtained at (16 x 16) block.

	DCT Block size				
	Size = 64 8 x 8	Size = 144 12 x 12	Size = 256 16 x 16	Size = 400 20 x 20	Size = 576 24 x 24
Detection rate (%)	76.8	76.6	77.7	76.5	76.4

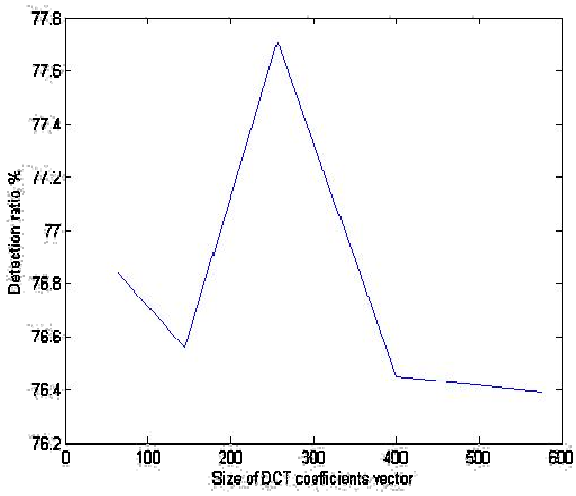


Figure 5.3: Detection rate versus size of feature vector. The best results were obtained for a feature space of dimensionality 256.

The third experiment compared the detection results of a hexagonal lattice topology to a rectangular lattice. Table 5.3 presents the results for both cases. Better results were obtained for the hexagonal lattice, presumably because each updating step affects a larger number of neighbours. This is also the reason that hexagonal lattices require shorter training times than rectangular lattices, as the update rule for more nodes leads to faster convergence.

TABLE 5.3
HEXAGONAL LATTICES VS. RECTANGULAR LATTICES

	Size =100 (10 × 10) Hexagonal	Size = 100 (10 × 10) Rectangular
Detection rate (%)	77.7 %	76.9 %
Total training time (seconds)	406.2	446.8

The previous experiments were not particularly concerned with computational requirements. Most of the computational load comes from the large size of the feature vectors being used, whereas the memory needs, derive primarily from the size of the SOM. Therefore the aim of fourth experiment was to determine whether a smaller feature vector could be constructed from a given set of DCT coefficients, without significant degradation in system performance.

A statistical analysis was conducted on a set of 3166 facial and non-facial skin samples. For the chosen DCT block size of 16 × 16, a total of 256 DCT coefficients were computed for each sample. Each of these coefficients can be viewed as a representation of a separate dimension in a 256-dimensional feature space. By assessing the variance in each dimension of this space, it is possible to determine which of the coefficients contribute most to the final decision of the classifier. Variances were computed using

$$\mathrm{var}\left(x_j\right) = \sum_{i=1}^{k} (X_i - \bar{X})^2 \qquad (5.4)$$

Where variable *j* is the DCT coefficient index, *i* is the sample index and *k* is equal to the available number of samples.

Figure 5.4 shows the statistical variances of the 256 features for this set of samples. The graph indicates that there are seven prominent peaks where high variance values occur. This suggests that these particular features perform prominent roles during classification. To exploit this, we defined a reduced-size feature space based on these high-variance DCT coefficients alone, whereas all other coefficients were excluded. The new feature vectors consist of only 27 DCT coefficients each. Note that this reduction in dimensionality is similar to a Karhunen-Loeve-type analysis.

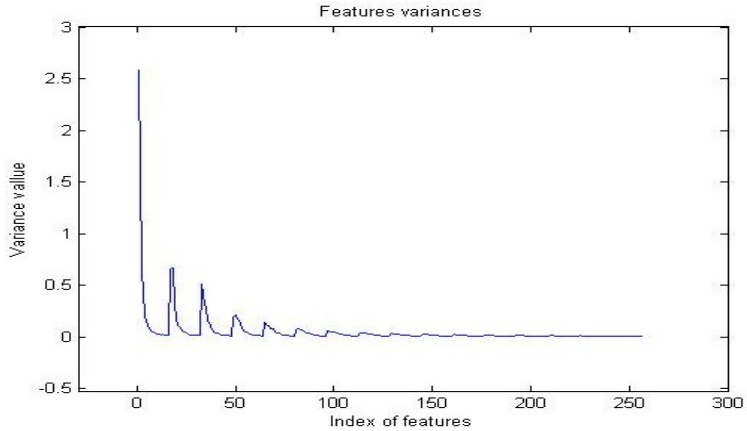


Figure 5.4: Statistical variances of the feature component values indicate seven significant cases of high-variance features.

Table 5.4 compares the performance of the system for full-size and reduced-size feature spaces. In spite of the dramatic reduction from 256 features to only 27, the detection rates are essentially the same. In addition to detection rates, the table also shows training times and memory usage for the both batch training and sequential training. The training length is kept constant, for fair comparison regarding the training time. The minimum value of the needed memory size *M* in the case of batch training is roughly estimated by

$$M = 8(5(m+n)d + 3m^2) \tag{5.5}$$

as described in [99], where *m* is the number of SOM neurons, *n* represents the number of samples, and *d* is the dimensionality of the input feature vector. The memory requirement is therefore proportional to the size of the feature vector.

This last experiment has demonstrated that good face detection performance is possible, even with feature vectors that are dramatically reduced in size relative to the usual case of DCT-based analysis. This makes the proposed method much more attractive for the low-cost, real-time implementation of a face-detection system. Commercial implementations of the SOM already exist [100]; thus it is conceivable that practical SOM-based face detection may be possible in the future.

TABLE 5.4
THE EFFECT OF REDUCING FEATURE VECTOR SIZE

DCT size	Batch training time (s)	Batch detection rate	Memory consumption (Byte)	Sequential training time ($s \times 10^3$)	Sequential detection rate
27	82.5	77.9%	1068240	1.0009	76.3 %
256	406.2	77.7 %	12712320	1.9070	77.6 %

5.3.2 Feature Extraction using 2D-DWT

In this section the 2D-DWT coefficients are calculated for each pre-cropped skin block resulting from the previous stage. This results in a matrix of 32 × 32 coefficients. A subset of these values is taken to construct the feature vector. Empirically, the upper right corner and the lower left corner of the 2D-DWT matrix contain the most important values, because they correspond to the vertical and horizontal components within the processed image block.

This section presents results for three experiments. The first experiment investigates which coefficients are to be used for constructing the feature vector. Table 5.5 shows the resulting detection rates, false positive rates and false negative rates for different combinations of DWT coefficients up to the first level, using the Haar wavelet transform. For more details on the corresponding locations of the DWT coefficients with respect to the original image coordinates, review Figure 4.5.

TABLE 5.5
2D-DWT COEFFICIENTS VS. DETECTION RATES

	Horizontal and Vertical details (512 Coeff.)	Horizontal, Vertical, and Diagonal details (768 Coeff.)	Approximated image, and Horizontal details (512 Coeff.)	Vertical and Diagonal details (512 Coeff.)
Detection rate (%)	69.95	70.94	63.55	67.00
False positive rate (%)	18.12	16.8	20.13	21.27
False negative rate (%)	30.05	29.06	36.45	33

The second experiment used a feature vector consisting of the vertical and horizontal details of the first wavelet decomposition level to compare the performance of the different wavelet families (Haar, Debauchi, Coiflet and Symmlet). Table 5.6 demonstrates the performance of each wavelet type in terms of detection rate, false positive rate and false negative rate. The table also shows the dimensionality of the feature vector for each wavelet family. The similarity between the acquired results using Haar and db1 matches the fundamental fact that they are two equivalent wavelets. The Haar wavelet achieved the highest detection rate; this can be justified

by it being less sensitive to noise. On the other hand, the rest of the wavelet types are highly sensitive to noise, as well as that they generate more correlated features.

TABLE 5.6
COEFFICIENTS OF 2D-DWT VS. DETECTION RATES

	Haar (512 Coeff.)	**db1** (512 Coeff.)	**db4** (722 Coeff.)	**db8** (1058 Coeff.)
Detection rate (%)	70	70	48.8	62.6
False positive rate (%)	18.1	18.1	29.3	28.1
False negative rate (%)	30	30	51.2	37.4
	Coif-1 (648 Coeff.)	**Coif-3** (1152 Coeff.)	**Sym4** (722 Coeff.)	**Sym8** (1058 Coeff.)
Detection rate (%)	68	65	67.5	61.6
False positive rate (%)	26.3	27.5	27.5	24.4
False negative rate (%)	32	35	32.5	38.4

The third experiment investigates the usage of additional vertical and horizontal details from farther levels of details. This is done using the Haar wavelet. Table 5.7 displays the summary of these results.

TABLE 5.7
PERFORMANCE ANALYSIS FOR USING DIFFETENT LEVELS OF HAAR WAVELET TRANSFORM

	Level 1 and 2 (256x2)+ (64x2) = 640 Coeff.		**Level 2 (64x2)= 128 Coeff.**	
	Size =64 (8 × 8)	Size = 100 (10 × 10)	Size =64 (8 × 8)	Size = 100 (10 × 10)
Detection rate (%)	73.2	74.9	73.9	71.9
False positive rate (%)	18.7	19.3	21.3	19.7
False negative rate (%)	26.6	25.1	26.1	28.1
	Level 1, 2 and 3 (256x2)+(64x2)+(16x2) =672 Coeff.		**Level 2, and 3 (64x2)+(16x2) =160 Coeff.**	
	Size =64 (8 × 8)	Size = 100 (10 × 10)	Size =64 (8 × 8)	Size = 100 (10 × 10)
Detection rate (%)	68.5	69.5	70.4	70.9
False positive rate (%)	17.8	20.7	20.8	22.9
False negative rate (%)	31.5	30.5	29.6	29

5.3.3 Feature Extraction using Geometrical Moment Invariants

In this section, we use a set of 29 features based on calculating the moments of each skin block up to the third order. Those features in use, include the mass of the block, a vector $\mathbf{I_1} = (\mu_{20}, \mu_{11}, \mu_{02}, \mu_{30}, \mu_{21}, \mu_{12}, \mu_{03})^T$ containing all the 2nd and 3rd order central moments, a vector $\mathbf{I}_2 = (H_1, H_2, H_3, H_4, H_5, H_6, H_7)^T$ whose length is seven includes the seven orthogonal and translational invariants given by Hu in (4.22), another 14 moments containing seven invariant moments to similitude transforms, as well as, seven invariant moments to similitude and orthogonal transforms. The feature vector $\mathbf{F}$ is defined as:

$$\mathbf{F} = (M, \mathbf{I_1}, \mathbf{I}_2, \mathbf{S_1}, \mathbf{S}_2)^T \tag{5.6}$$

This section presents the results of two experiments. The first experiment examines the use of the feature vector $\mathbf{F}$. Table 5.8 shows the results for two different sizes of SOM classifier using a hexagonal lattice grid structure.

TABLE 5.8
SOM SIZE VS. DETECTION RESULTS

	Size =64 (8 × 8)	Size = 100 (10 × 10)
Detection rate (%)	62.6	67
False positive rate (%)	26.5	26.5
False negative rate (%)	37.4	33

The second experiment investigates the performance after reducing the dimensionality of the feature vector to 12 using PCA. Table 5.9 shows the significant improvement gained over the previous experiment. The improvement in performance of PCA can be justified by the fact that PCA succeeded to minimize the correlation among the features. The previously mentioned experiments are implemented using the toolbox for the Lifting Scheme on Quincunx Grids (LISQ) [101].

TABLE 5.9
THE EFFECT OF REDUCING FEATURE VECTOR SIZE USING PCA

	Size =64 (8 × 8)	Size = 100 (10 × 10)
Detection rate (%)	71.9	73.9
False positive rate (%)	29.2	29.7
False negative rate (%)	28.1	26.1

5.3.4 Feature Extraction using Edge Detection

In this section, we construct a set of features from the binary block that is produced as an output from the edge detection process, using the Canny algorithm. In the first experiment, a set of 66 features is used. These 66 features are assembled as follows: 32 values represent the sum of edge pixels per each row normalized by the number of columns, 32 values represent the sum of edge pixels per each column normalized by the number of rows, and 2 values represent the sum of edge pixels per both diagonals also normalized by the number of pixels per diagonal. For a block size of (32 × 32), the normalization factor is 32. Another set of features is used in the second experiment; this set is constructed by calculating the 29 features based on the geometrical moments that are introduced in sub-section 5.3.3 of this thesis. The main difference is that the geometrical moments are not calculated for the intensity values of the skin block but they are calculated for the binary block that results from the edge detection process. Then, the dimensionality of the moments feature vector is reduced into only 12 dimensions using PCA. This set of features is called moments of edges.

This section presents the experimental results for the two previously mentioned experiments. Table 5.10 below, shows the effect of different SOM sizes when using the first set of edge features. Furthermore, table 5.11 shows the effect of using the same SOM sizes when using moments of edges for feature extraction.

TABLE 5.10
SOM SIZE VS. DETECTION RESULTS OF CANNY EDGE DETECTION ALGORITHM USING 66 FEATURES

	Size =16 (4 × 4)	Size = 64 (8 × 8)	Size =100 (10 × 10)	Size = 121 (11 × 11)
Detection rate (%)	70	75.4	75.9	75.4
False positive rate (%)	18.4	20	20.1	21.7
False negative rate (%)	30.1	24.6	24.1	24.6

TABLE 5.11
SOM SIZE VS. DETECTION RESULTS OF MOMENTS OF EDGES USING 12 MOMENTS

	Size =16 (4 × 4)	Size = 64 (8 × 8)	Size =100 (10 × 10)	Size = 121 (11 × 11)
Detection rate (%)	76.9	85.2	78.8	77.3
False positive rate (%)	18.5	32.8	26.4	33.1
False negative rate (%)	23.2	14.9	21.2	22.7

5.3.5 Feature Extraction based on fusion of DWT features and geometrical moments

This section introduces a novel scheme to construct a hybrid feature vector that merges features that are extracted using two different feature extraction methods. The hybrid feature vector is constructed by concatenating a set of selected DWT coefficients with a set of geometrical moments. The set of DWT coefficients are selected based on the detection results are shown in subsection 5.3.2. The results show that the best detection results are given when using the Haar wavelet up to the second level. In this case, the feature vector consists of the vertical and horizontal details given by the two levels Haar decomposition. This gives a feature vector whose size is 640. Concurrently, the geometrical moments that are used, are the same 12 moments that are selected by the PCA test in experiment 2 in subsection 5.3.3

In order to get the highest possible detection rate, pre-analysis is carried out on the two feature vectors before merging. Firstly, a PCA test is carried out on the involved set of Haar coefficients to select a smaller subset whose size is p while keeping a convenient detection result. The aim of this analysis is to find out which features and their minimum number that will give us the best detection results. Repeatedly, the analysis determined that the best detection results were gained at p=45. Figure 5.5 demonstrates the relation between the detection rate and the size of the DCT based feature vector. The highest performance is clearly noticeable at p=45. Figure 5.6 shows the receiver operating characteristic (ROC) curve for different sizes of the SOM. This curve was obtained by changing the size of the feature vector while maintaining the SOM size and grid structure.

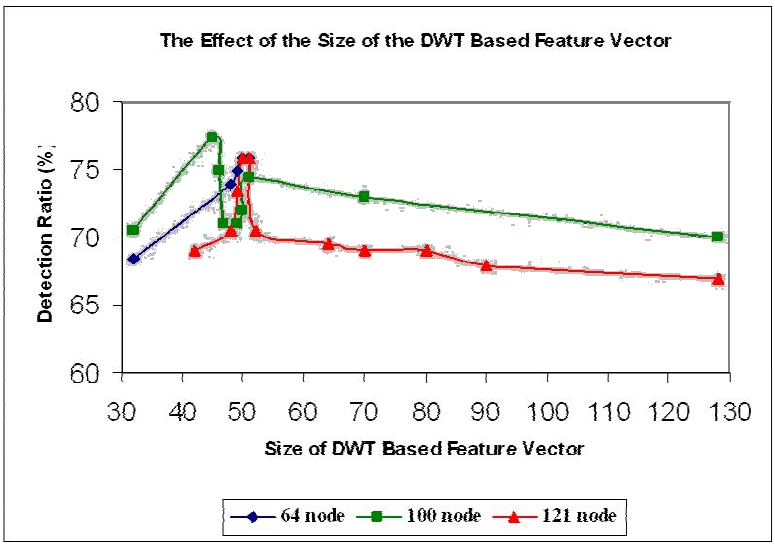


Figure 5.5: The effect of the size of the DCT based feature vector.

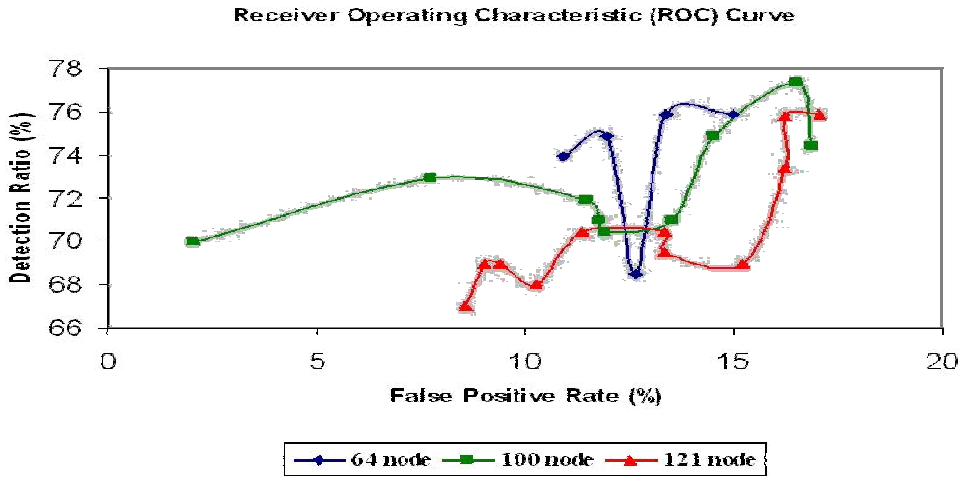


Figure 5.6: Receiver Operating Characteristic (ROC) curves for SOM sizes of 64, 100 and 121 nodes respectively.

Finally, the hybrid feature vector is constructed by merging the 45 DWT coefficients with the 12 geometrical moment invariants. The result is a feature vector whose size is 57 features. Table 5.12 summarizes the detection results that were obtained using this hybrid feature vector.

TABLE 5.12
SOM SIZE VS. DETECTION RESULTS OF THE HYBRID FEATURE VECTOR
(45 DWT COEFFICIENTS + 12 GEOMETRICAL MOMENTS= 57 FEATURES)

	Size =16 (4 × 4)	Size = 64 (8 × 8)	Size =100 (10 × 10)	Size = 121 (11 × 11)
Detection rate (%)	74.9	84.2	88.7	82.3
False positive rate (%)	9.2	14.3	21.4	18.9
False negative rate (%)	25.1	15.8	11.3	17.7

5.3.6 Feature Extraction based on fusion of DCT features and Edges information

This section presents another scheme for constructing the hybrid feature vector using two different feature extraction methods. It depends on the same idea presented in the previous section, but other feature extraction methods are employed. In this scheme, the hybrid vector is constructed by concatenating a set of selected DCT coefficients with a set of edge information that is obtained by canny edge detection. In this section, the results of two experiments are proposed. In the first experiment the edge information is given by the 66 features previously explained in sub-section 5.3.5, where in the second experiment the edge information is given by the moments of edges that are described in the same prior sub-section

Pre-analysis and simplification are needed before the merging process. The number of DCT coefficients is 1024 when applying a DCT on a skin block of size 32×32. This is a very large number for a dimension of a feature vector, especially in terms of memory consumption for hardware implementations. Figure 5.7 demonstrates the relation between the detection rate and the size of the DCT based feature vector. The scale on the horizontal axis is normalized by a factor of 4. Each step on the x-axis is equivalent to adding four more features to the feature space. A PCA test was used to reduce the dimensionality of the DCT coefficients to p features. Repeating the test yielded the best average of detection results at p=56 .

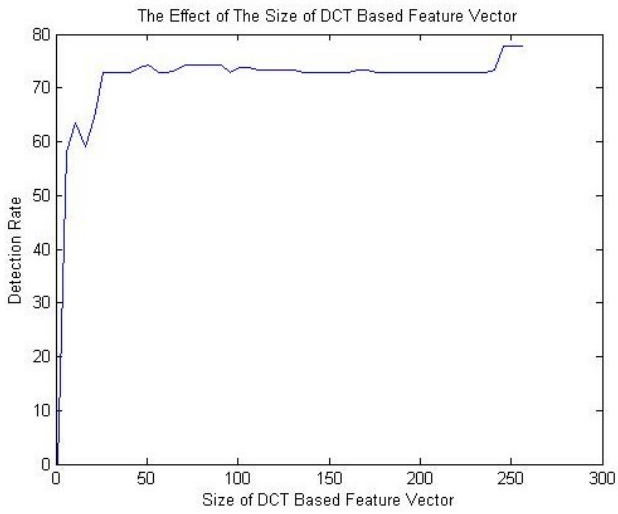


Figure 5.7: The effect of the size of the DCT based feature vector.

Figure 5.8 shows the receiver operating characteristic (ROC) curve for different sizes of the SOM. This curve was acquired by changing the size of the feature vector while maintaining the SOM size and grid structure.

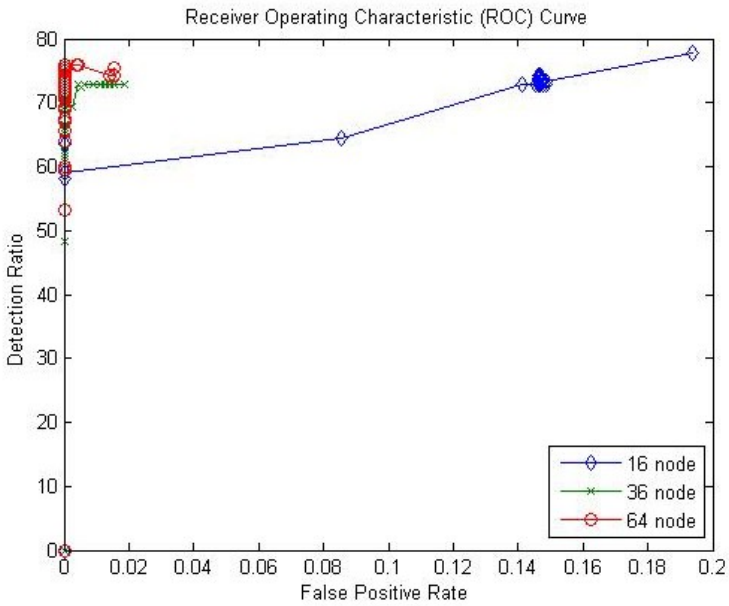


Figure 5.8: Receiver Operating Characteristic (ROC) curves for SOM sizes of 16, 36 and 64 nodes respectively.

For the sake of dimensionality reduction of the edge based feature vector, a simple variance analysis was carried out on the first set of Canny based features which includes 66 features as an effort to select only 32 features. Figure 5.9 shows a plot of these 66 variance values before choosing the features that are corresponding to the highest 32.

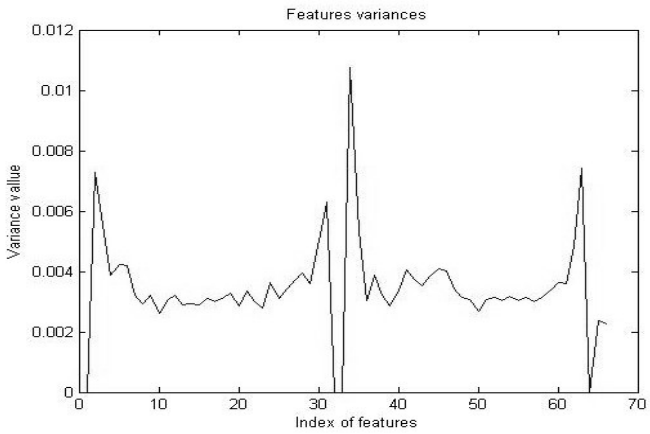


Figure 5.9: Statistical variance test determines 32 maximum values for the Canny edge-based features.

At the end of the pre-analysis, the 56 DCT based features are merged with the 32 Canny based features to construct the hybrid feature vector. This builds a hybrid feature vector whose size is 88. This is the feature construction scheme used within the first experiment in this section. Table 5.13 below shows the detection results when using this hybrid feature vector.

The second experiment constructed the hybrid feature vector by merging the 56 DCT Coefficients and the 12 moments of edges. Those moments of edges are the same moments that were selected by the PCA test in the second experiment in

subsection 5.3.4. This yielded a feature vector whose size is only 68. Table 5.14 summarizes the results of this experiment.

TABLE 5.13
SOM SIZE VS. DETECTION RESULTS OF THE FIRST HYBRID FEATURE VECTOR
(56 DCT COEFFICIENTS + 32 CANNY EDGE BASED FEATURES = 88 FEATURES)

	Size =16 (4 × 4)	Size = 64 (8 × 8)	Size =100 (10 × 10)	Size = 121 (11 × 11)
Detection rate (%)	65	89.2	82.3	80.3
False positive rate (%)	16	13.7	17.5	17.2
False negative rate (%)	35	10.9	17.7	19.7

TABLE 5.14
SOM SIZE VS. DETECTION RESULTS OF THE SECOND HYBRID FEATURE VECTOR
(56 DCT COEFFICIENTS + 12 MOMENTS OF EDGES = 68 FEATURES)

	Size =16 (4 × 4)	Size = 64 (8 × 8)	Size =100 (10 × 10)	Size = 121 (11 × 11)
Detection rate (%)	77.7	87.4	81.2	79.9
False positive rate (%)	13.5	13.8	19.4	18.1
False negative rate (%)	22.4	12.6	18.8	20.1

5.3.7 Voting based detection using multiple SOM

This section presents a novel idea of making the classification decision based on combining multiple decisions from multiple classifiers. An inverted exclusive OR operation is applied on the decisions that are made by three SOM based classifiers to make a final decision. It is the same as selecting the best two out of three. A special case is handled separately when all the three pre-decisions are either positive or negative. This idea is different from the idea of the fusion of features. Here the decisions are merged using the voting scheme that is shown in figure 5.11 where each classifier is trained using a pure set of features. On the other hand the previous idea that was presented in subsection 5.3.5 and 5.3.6 is dependent on training a single classifier using a hybrid set of features as shown in figure 5.10.

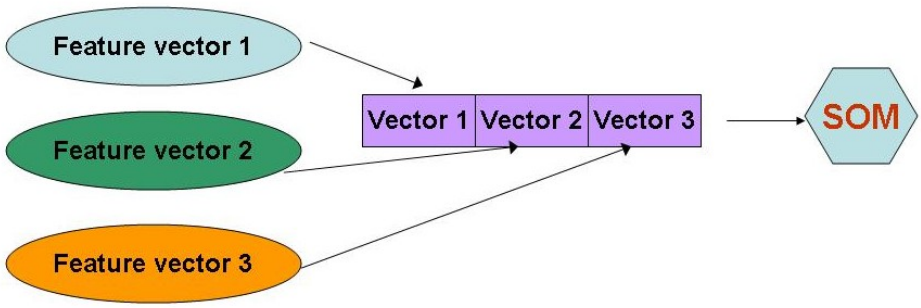


Figure 5.10: One SOM is used for training and testing using two or more feature vectors that are concatenated to form one feature vector.

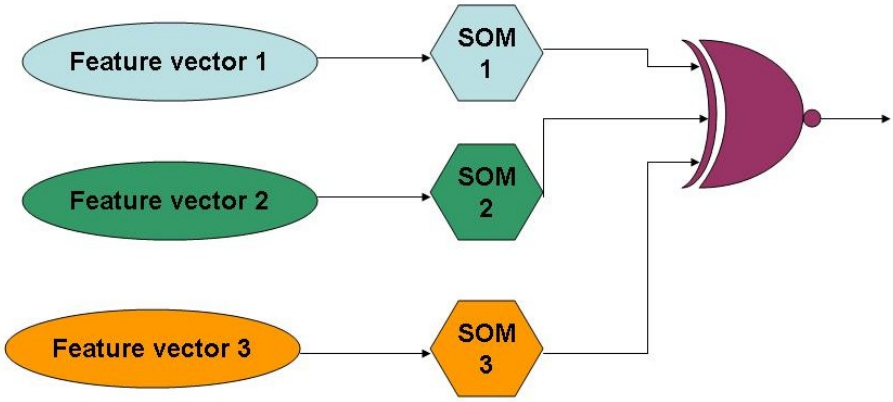


Figure 5.11: Voting Scheme using multiple SOM networks; each one is trained using a different family of features.

In this experiment, voting was done using three SOM based classifiers as shown in figure 5.11. Every SOM classifier is trained using a set of feature vectors that is extracted using a single feature extraction method. The experiment involved training the first SOM using edge based features. It used the same 66 features that were used in the first experiment in subsection 5.3.4. The second SOM classifier was trained using the 45 uncorrelated DWT coefficients that were used to build the hybrid vector in subsection 5.3.5. Finally, the last SOM classifier was trained using the 12 geometrical moments that were selected by the PCA test in subsection 5.3.3. The detection results of this experiment are summarized in table 5.15.

TABLE 5.15
DETECTION RESULTS USING VOTING BASED CLASSIFICATION
(66 EDGE BASED FEATURES + 45 DWT COEFFICIENTS + 12 MOMENTS OF EDGES = 123 FEATURES)

	Size =16 (4 × 4)	Size = 64 (8 × 8)	Size =100 (10 × 10)	Size = 121 (11 × 11)
Detection rate (%)	62.2	79.4	75.4	70.8
False positive rate (%)	7.2	15.1	13.5	12.8
False negative rate (%)	37.8	20.6	24.6	29.2

CHAPTER 6
FPGA Based Framework for Hardware Implementation

This chapter presents a framework for an FPGA based hardware implementation of a real-time face detection system. Realizing the presented framework will implement the face detection techniques that are presented in the earlier chapters of this thesis. This chapter also illustrates the implementation of a part of the system where a real-time skin segmentation unit is realized.

6.1 An FPGA based framework

To the best of our knowledge, only two complete real-time face detection hardware implementations have been proposed in the literature. Those include the system implemented by McCready [102] and the system implemented by Theocharides [103]. Both operate on real-time video data. The first uses 16 FPGA boards, and each is an Altera 10K100. It runs at a system clock frequency of 12.5 MHz processing 30 frames/ second. The second system was prototyped using a Xilinx XUP2V-Pro development board, which has a Xilinx XC2VP30 FPGA on board. The system operates at a system clock frequency of 125 MHz, processing 24 frames per second at a grayscale resolution of 320×240.

In this thesis, we propose a new FPGA based framework for a face detection system that is running in real-time. The framework uses a Xilinx XUP2V-Pro development board as a base for hardware implementation. A block diagram of the system is shown in figure 6.1.

The tasks of the video channel are done by the VDEC1 Video Decoder Board [104]. It has an Analog Devices ADV7183B decoder chip. It automatically detects and converts the standard analog video signals, such as NTSC, SECAM, and PAL. The output is a digital video stream in YCbCr format with sampling rate of 4:2:2. After converting the analog signal into a digital stream, it is time to start the face detection operation.

The face detection system is partitioned primarily into three units. The face detection operation starts with the pre-processing unit that is responsible for the following tasks: up-sampling, conversion of YCbCr space into RGB space, skin segmentation, conversion of RGB into greyscale, morphological opening, and labelling of connected components. A separate block diagram for the pipelined internal stages of this unit is shown in figure 6.2.

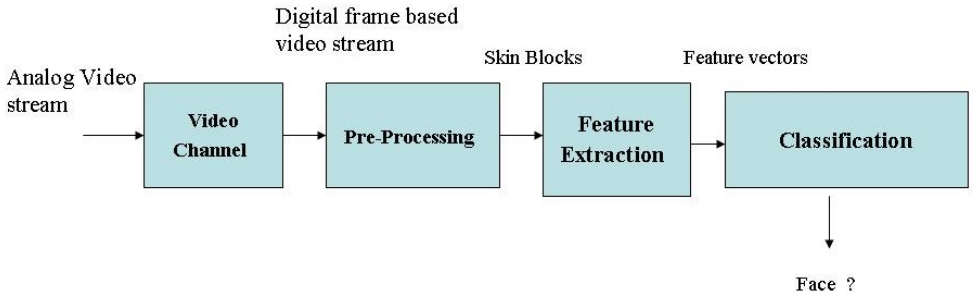


Figure 6.1: A block diagram for the real-time face detection system.

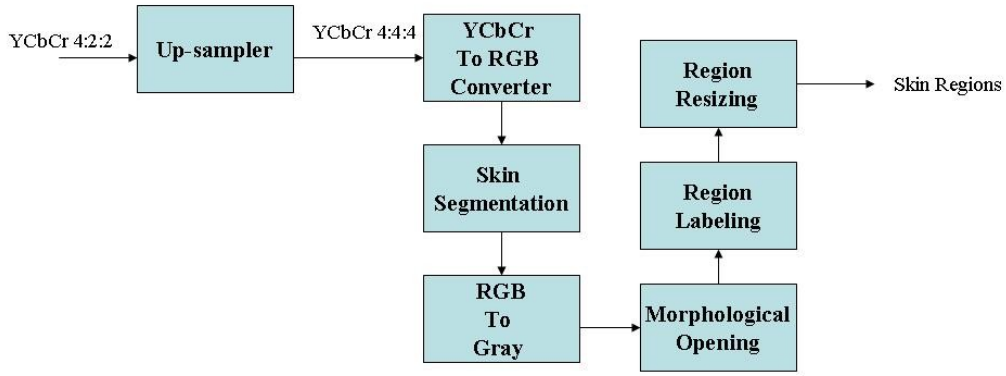


Figure 6.2: A block diagram for the pre-processing unit of the hardware framework

The second main unit is concerned with constructing the feature vector of every skin region that comes out of the previous unit. The internal structure of the feature extraction unit, changes according to which method will be used and the nature of its algorithm. For example, the inherent parallelism found in the discrete cosine and discrete wavelet transforms, suggests the usage of a parallel implementation. This is expected to enhance the whole performance of the system.

Many different real-time hardware implementations can be created for the feature extraction methods that are presented in this thesis. Regarding the discrete cosine transform, several FPGA implementations have been proposed [72, 73, 77]. As for the discrete wavelet transform, the various implementations that exist vary in terms of processing speed, power consumption, as well as memory usage [74, 75, 76, 105, 106]. Real-time implementations of geometrical moments and moment invariants require intensive computation needs and memory consumption. Several attempts addressed this problem, presenting several implementations to calculate the moments of an image in real-time [107, 108, 109, 110]. Finally, for edge detection, different edge detection algorithms have been implemented using FPGA technology, including the Sobel and Canny algorithms [111,112,113].

The last unit in the presented framework is the unit that realizes the learning based classification. As described earlier, it uses an SOM based classifier. Recently, few hardware implementations of SOM based classifier using FPGA technology have been proposed [114,115]. The AXEOM learning processor that is used in [100] to implement a real-time skin detector is a parallel processor with a neuron array capable of one million classifications per second with a moderate clock speed. Studying the possibility of integrating the pre-processing and feature extraction units with the AXEOM learning processor, may be a perfect option toward implementing of a fully running real-time system with outstanding performance.

6.2 Implementation of real-time skin segmentation unit

This section illustrates the implementation of a part of the presented hardware framework. The partial implementation includes the pre-processing stage up to the skin segmentation block as shown in figure 6.2, so that it is called an implementation of a real-time skin segmentation unit.

This small scale implementation represents a case study for the hardware implementation of the face detection system. First, the functionality of the skin segmentation unit was tested using the Matlab computational environment [28], Simulink [118] and Xilinx System Generator for DSP [119]. Figure 6.3 shows the design block diagram in Simulink before generating the module described by VHDL or Verilog. The skin segmentation VHDL module was generated using Xilinx System Generator and then simulated by the hardware software co-design feature found in Xilinx System Generator v8.1. Xilinx System Generator supports the hardware in-loop simulation capability where the output of the hardware module running on FPGA is forwarded to the Matlab environment to be analyzed. This is done through a USB communication channel with the Xilinx XUP2V-Pro development board. It operates on frequency of 12 MHz. Figure 6.4 shows an example of the input and output for a particular simulation.

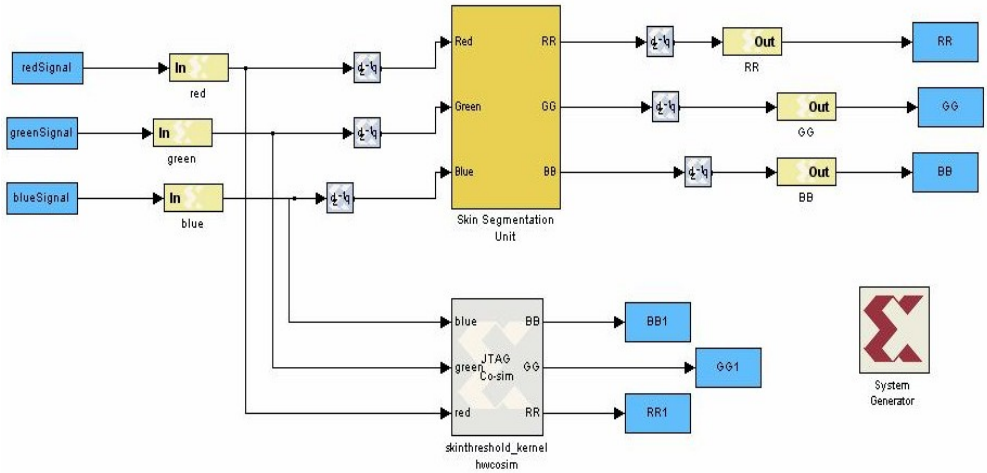


Figure 6.3: Skin segmentation unit: Simulink design block diagram.

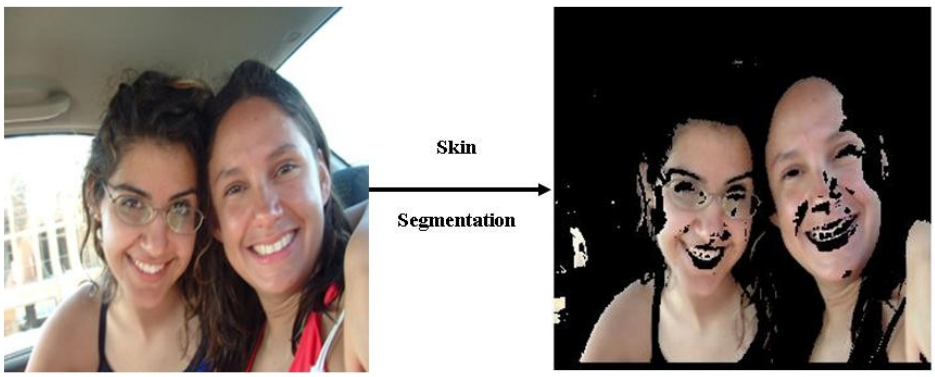


Figure 6.4: Example of skin segmentation using hardware software co-design features in Matlab, Simulink, and System Generator for DSP.

As for the real-time implementation, an existing library for processing of a video input is used to facilitate processing of the input video stream. The library is compiled and distributed by Xilinx [118]. The library includes the required modules for receiving and forwarding a video stream from the VDEC1 video decoder board to the SVGA port that is found on the XUP board. We use this library to select one among different sources of the video input, as well as to display the output of the skin segmentation unit. The library includes the module that is responsible for up-sampling the YCrCb digital stream with a 4:2:2 sampling rate to a YCrCb digital stream with a 4:4:4 sampling rate. It also offers a module that converts the three Y, Cr, and Cb components, where each value is represented by ten bits, into red, green, and blue components where each value is represented by one byte.

The main skin segmentation module, whose task is to segment the input video stream frame by frame based on the skin color, is coded using the Verilog hardware description language. The module was coded, compiled, and synthesized using Xilinx ISE Foundation 8.1i [120] and it has been integrated with the Xilinx video processing library using Xilinx Platform Studio 8.1i [121]. This module inherits three other new modules. The first module is designed to calculate the maximum value of two input values. The second module calculates the minimum value of two input values of the same size. And finally the last one calculates the absolute value of the difference between two input values. These modules are integrated in order to implement the equation of the comparison based segmentation that is given by (4.1). The module hierarchy is shown in figure 6.5.

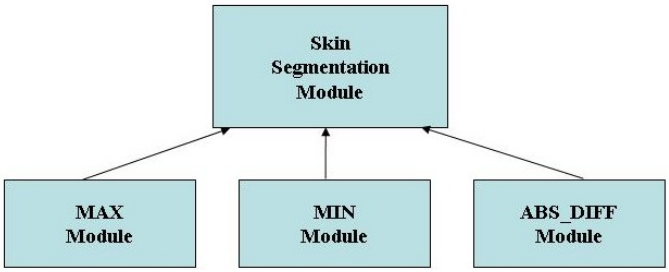


Figure 6.5: The hierarchy of the skin segmentation module.

The whole system, including the video up-sampling module, the color space conversion module, the skin segmentation module and SVGA display module, has been tested after integration to operate at a frequency clock of 100 MHz. It has been able to process 30 color frames per second at a resolution of 1024×768. The skin segmentation module separately is able to process up to 271 frames per second. The difference is the processing delay because of other parts of the system. Figure 6.6 shows the integrated hardware during testing and running.

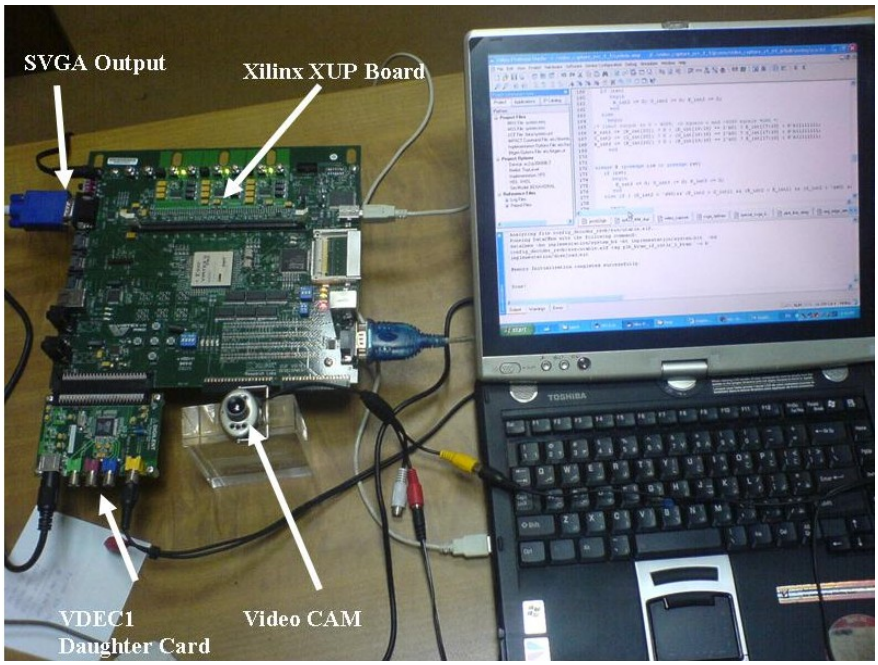


Figure 6.6: The skin segmentation unit.

CHAPTER 7
Conclusion and Future Work

7.1 Summary and Conclusion

This thesis has introduced the face detection problem as a pattern recognition problem of three stages. These stages are pre-processing, feature extraction, and learning based classification. New methods and techniques have been presented, tested and evaluated for each stage. These techniques have been chosen such that they could be implemented in today's hardware for real-time operation.

A new color image database has been compiled, and made available online for researchers. It is highly adequate for testing and evaluation of face detection algorithms and skin segmentation techniques. It presents test cases that have real challenges for an algorithm under testing. It is called the VT-AAST database.

A new approach for pre-processing, including skin segmentation, morphological opening, and labeling of connected components, has been used to determine the candidate regions that are most likely to constitute human faces. It has the advantage of the very low processing requirements compared to the traditional approaches that are dependent on the comprehensive search within the image, such as the generation of the multi-resolution pyramid of the image.

Several feature extraction methods were investigated in order to achieve a reasonable detection results in terms of detection rate and false positive rate. These include the two dimensional discrete cosine transform (2D-DCT), the two dimensional discrete wavelet transform (2D-DWT), edge detection, and geometrical moment invariants. In addition, different combinations of these methods were also considered for constructing hybrid feature vectors.

As for the learning based classification stage; we used a self organizing map neural network (SOM-NN) for classification. Various structures of the SOM were used. The effect of different SOM parameters was studied including the size, the initialization parameters, the number of neighbors, and the grid topology. On top of that, a novel paradigm was proposed for merging the decisions of multiple SOM based classifiers to obtain reasonable final detection rate at higher certainty of the results (low false positive rate).

The experimental results are listed in detail with performance analysis in chapter 5. The results are very promising. The highest detection rate that was achieved is 89.2% at a false positive rate of 13.7%. This is considered a very good result taking into account the high level of challenges that presented by the VT-AAST database.

Finally, since all the early mentioned techniques are implement-able in real-time hardware. An FPGA based implementation framework was introduced that can be carried out to implement a complete real-time face detection system. Implementation of skin segmentation pre-processing unit is presented as an example for hardware implementation.

7.2 Future Work

A possible extension for this work is to try the usage of other families of moment invariants for feature extraction. Suggestions include Zernike moments, Flusser moments and affine moment invariants (AMIs). The complexity of these moments and the possibility of hardware implementation may be studied.

In addition, preprocessing stage using traditional image scanning techniques such as the generation of the image multi-resolution pyramid can be combined with the proposed feature extraction and SOM classifier to design an offline implementation of the proposed face detection technique. Such system may perform better because it doesn't suffer of any missed candidate regions due to malfunctioning of the skin segmentation based preprocessing stage.

Finally, it should be possible to continue implementing the rest of the hardware framework to end with a complete FPGA based face detection prototype running in real-time as an embedded system. Further more, faster implementations can be achieved by optimizing the hardware design of each stage separately by obtaining optimized implementations for the calculation of the DCT, DWT, edge detection, or geometrical moments. This is can be done taking into account the parallelism nature of these algorithms that may help utilize the available hardware resources.

References

[1] M. H. Yang, and D. J. Kriegman, "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, Jan. 2002, pp. 34 - 58.

[2] E. Hjelmås, and B. K. Low, "Face Detection: A Survey", *Computer Vision and Image Understanding*, Vol. 83, No. 3, Sept. 2001, pp. 236-274.

[3] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, Jan. 1998, pp. 23-38.

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

[5] H. Schneiderman, and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2000, pp. 746-751.

[6] H. Kruppa, M. A. Bauer, and B. Schiele, "Skin Patch Detection in Real-World Images", *Proc. of the DAGM-Symposium*, 2002, pp. 109-116.

[7] P. Viola and M. Jones., "Robust Real-Time Face Detection", *International Journal of Computer Vision (IJCV)*, 2004, pp. 137-154.

[8] T. Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin, and W. Wolf, "Embedded Hardware Face Detection", *Proc. 17th International Conference on VLSI Design*, Jan. 2004, p.133.

[9] A. C. Loui, C. N. Judice, and S. Liu, "An Image Database for Benchmarking of Automatic Face Detection and Recognition Algorithms", *Proc. of IEEE International Conference on Image Processing (ICIP)*, Vol. 1, 1998, pp.146-150.

[10] P. Sharma, and R.B. Reilly, "A Color Face Image Database for Benchmarking of Automatic Facial Detection Algorithms", *Proc. 4th European Conference of Video/Image Processing and Multimedia Communications*, July 2003, pp, 423 - 428.

[11] P. J. Phillips, P. Rauss, and S. Der, "FERET (Face Recognition Technology) Recognition Algorithm Development and Test Report", *Technical Report ARL-TR 995, U.S. Army Research Laboratory*, October 1996.

[12] K-K. Sung, and T. Poggio, "Example-Based Learning for View-based Human Face Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, Jan. 1998, pp. 39-51.

[13] A. Martinez, and R. Benavente, "The AR Face Database," *Technical Report CVC 24, Purdue University*, 1998.

[14] R-L. Hsu, and M. Abdel-Mottaleb, "Face Detection in Color Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, May 2002, pp. 696 - 706.

[15] C. Garcia, and G. Tziritas, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis", *IEEE Transactions on Multimedia*, Vol. 1, No. 3, September 1999, pp. 264 - 277.

[16] C. Lin, "Face Detection by Color and Multilayer Feedforward Neural Network", *Proc. IEEE International Conference on Information Acquisition*, June-July 2005, pp. 518-523.

[17] K. Sandeep and A.N. Rajagopalan, "Human Face Detection in Cluttered Color Images using Skin Color and Edge Information", *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*, Dec. 2002.

[18] U.S. Census Bureau, 2000 Census of Population, Public Law 94-171 http://factfinder.census.gov/.

[19] Y. Wei, X. Bing, and C. Chareonsak, "FPGA Implementation of ADABOOST Algorithm for Detection of Face Biometrics", *Proc. IEEE International Workshop on Biomedical Circuits and Systems*, Dec. 2004, pp. S1/6.17-20.

[20] E. Osuna, R. Freund, and F. Girosit, "Training Support Vector Machines: An Application to Face Detection", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1997, pp. 130–136.

[21] Y. Li, S. Gong, and H. Liddell, "Support Vector Regression and Classification-based Multi-view Face Detection and Recognition", *Proc. 4th IEEE International Conference on Automatic Face and Gesture Recognition*, March 2000, pp. 300–305.

[22] Adobe Photoshop CS2, http://www.adobe.com/products/photoshop/.

[23] O. Bernier, M. Collobert, R. Feraud, V. Lemaire, J. E. Viallet, and D. Collobert, "MULTRAK: A System for Automatic Multiperson Localization and Tracking in Real-time", *Proc. International Conference on Image Processing, ICIP 98*. Vol. 1, Oct. 1998, pp. 136–140.

[24] S.Z. Li and Z. Zhang, "FloatBoost Learning and Statistical Face Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, Issue No. 9, Sept. 2004, pp. 1112-1123.

[25] S.Z. Li, L. Zhu, Z. Zhang, and H.-J. Zhang, "Learning to Detect Multi-view Faces in Real-time", *Proc. 2nd International Conference on Development and Learning*, June 2002, pp. 172-177.

[26] Microsoft Office Excel, http://office.microsoft.com/en-us/excel/default.aspx.

[27] Microsoft Office Access,http://office.microsoft.com/en-us/access/default.aspx.

[28] Matlab Online, http://www.mathworks.com/products/matlab/.

[29] T. Theocharides, N. Vijaykrishnan, and M. J. Irwin, "A Parallel Architecture for Hardware Face Detection", *Proc. IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, March 2006.

[30] Computer Vision laboratory, Faculty of Computer and Information Science, University of Ljubljana, Slovenia, http://www.lrv.fri.uni-lj.si/facedb.html.

[31] E. Marszalec, B. Martinkauppi, M. Soriano, M. Pietikäinen (2000), "A Physics-Based Face Database for Color Research", *Journal of Electronic Imaging*, Vol. 9 No. 1 pp. 32-38.

[32] Vladimir V., V. Sazonov, and Alla A.,"A Survey On Pixel-Based Skin Color Detection Techniques", *Proc. of Graphicon conference*,2003, pp. 85-92.

[33] Peer, P., Kovac, J., And Solina, "Human Skin Color Clustering for Face Detection", *EUROCON International Conference on Computer as Tool*, 2003, pp. 144-148.

[34] Kruppa, H., Bauer, M. A., And Schiele, "Skin Patch Detection in Real-World Images", *Proc. Annual Symposium for Pattern Recognition of the DAGM*, 2002, Springer LNCS 2449, pp.109–117.

[35] Yang, M.-H., And Ahuja, "Detecting Human Faces in Color Images", *International Conference on Image Processing (ICIP)*, 1998, vol.1, pp.127–130.

[36] Jedynak, B., Zheng, H., Daoudi, M., And Barret, ”Maximum Entropy Models for Skin Detection”, *Tech. Rep. XIII, Universite des Sciences et Technologies de Lille*, France, 2002.

[37] R. C. Gonzalez and R. E. Woods. *Digital Image Processing* Prentice Hall, 2002.

[38] Gardner Richard J. "The Brunn-Minkowski Inequality", *Bull. Amer. Math. Soc.*, 2002, pp. 355–405.

[39] Emmanuel Ifeachor and Barrie Jervis. *Digital Signal Processing: A Practical Approach*. Prentice Hall, 2001.

[40] Ma, L., Xiao, Y., Khorasani, K., and Ward, “A New Facial Expression Recognition Technique Using 2D DCT And K-Means Algorithm”, *International Conference on Image Processing (ICIP), 2004*, Volume 2, pp.1269 – 1272.

[41] L. Ma and K. Khorasani, ” Facial Expression Recognition Using Constructive Feedforward Neural Networks”, *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Volume 34, Issue 3, June 2004, pp.1588 – 1595.

[42] S. A. Khayam, “The Discrete Cosine Transform (DCT): Theory and Application”, *Department of Electrical & Computer Engineering, Michigan State University*, Tutorial, March 10th 2003.

[43] K. Masselos, Y. Andreopoulos, and T. Stouraitis, “Performance Comparison Of Two-Dimensional Discrete Wavelet Transform Computation Schedules On A VLIW Digital Signal Processor”, *IEE Proc. Vis. Image Signal Process.*, Vol. 153, No. 2, April 2006, pp.173-180.

[44] I. Daubechies, “Ten Lectures on Wavelets”, *Society for Industrial and Applied Mathematics*, 1992, ISBN 0-89871-274-2.

[45] J. S. Walker. *A Primer On WAVELETS And Their Scientific Applications*. CHAPMAN & HALL/CRC, 1999.

[46] A. Graps, “An Introduction to Wavelets”, *IEEE Computer Science And Engineering*, Vol. 2, Num. 2, 1995.

[47] A. Teolis. *Computational Signal Processing with wavelets.* Birkhauser, 1998.

[48] A. Akansu and R. Hadad. *Multiresolution Signal Desomposition: Transforms, Sub-Bands And Wavelets*. Academic Press/Harcourt Brace Jovanovich, 1992.

[49] M. K. Hu, “Visual Pattern Recognition by Moment Invariants”, *IEEE Transaction on Information Theory*, No. 8, 1962, pp. 179-187.

[50] A. Abu-Zaid, O. Hinton, and E. Horne, “About Moment Normalization And Complex Moment Descriptors”, *Proc. International Conference on Pattern Recognition 88, IEEE Comp. Press*, pp.399- 407.

[51] Y. S. Abu-mostafa and D. Psaltis, “Recognitive Aspects of Moments Invariants“, *IEEE Transaction on Pattern Analysis and Machine Intelligence 6*, 1984, pp.698-706.

[52] D. Slatar and G. Healey, “The Illumination-Invariant Recognition Of 3D Objects Using Color Invariants”, *IEEE Transaction on Pattern Analysis and Machine Intelligence 18*, 1996, pp.206-210.

[53] F. Mindru, T. Moons and L. Van Gool, “Recognizing Color Pattern Irrespective of Viewpoint and Illumination”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp.368-373.

[54] J. Flusser, “On The Independence of Rotation Moment Invariant”, *the journal of the Pattern Recognition society, Vol. 33*, 2000, pp.1405-1410.

[55] R. Bidoggia and S. Gentili, “A Basis of Invariant Moments for Color Images”, *Proc. of 9th International Workshop on Systems, Signal and Image Processing (IWSSIP'02)*, Manchester, 7-8 November 2002, pp.527-531.

[56] Gouda I. Salama and A. Lynn Abbott, ”Moment Invariants And Quantization Error”, *IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 157-163.

[57] Kim W., Sung, ”A Region Based Shape Descriptor Using Zernike Moments”, *16th Conference on Signal Processing and Image Communication*, 2000, pp.95-102.

[58] Canny, J., ”A Computational Approach to Edge Detection”, *the 8th IEEE Trans. Pattern Analysis and Machine Intelligence*, 1986, pp.679-714.

[59] Lindeberg, T., "Edge Detection And Ridge Detection With Automatic Scale Selection", *International Journal of Computer Vision*, 30, 2, 1998, pp.117-154.

[60] Ziou, D. and Tabbone, “Edge Detection Techniques: An Overview”, *International Journal of Pattern Recognition and Image Analysis*”, 1998, pp. 537-559.

[61] I.K. Fodor, “A Survey of Dimension Reduction Technique”, *UCRL-ID-148494, U.S. Department of Energy*, May 9, 2002.

[62] K.V. Mardia, .J.T. Kent, and J.M. Bibby. *Multivariate Analysis. Probability and Mathematical Statistics*. Academic Press, 1995.

[63] Haykin Simon. *Neural Networks-A Comprehensive Foundation*. 2nd Edition. Prentice-Hall, 1999.

[64] Teuvo Kohonen. Self-Organizing Maps, *Volume 3- of Springer Series in Information Science*. Springer, Berlin, Heidelberg, 1995.

[65] T. Kohonen, “Self-organized Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, 43, 1982, pp.59-69.

[66] T.Kohonen,“The Self-Organizing Map”, *Proceedings of the IEEE,78(9)*, 1990,pp.1464-1480.

[67] J. Lampinen, and E. Oja, “Clustering Properties of Hierarchical Self-Organizing Maps”, *Journal of Mathematical Imaging and Vision*, 1992,pp. 261- 272.

[68] Jason Ong and S. S. R. Abidi, “Data Mining Using Self-Organizing Kohonen Maps: A Technique For Effective Data Clustering & Visualisation”, *International Conference on Artificial Intelligence (IC-AI'99)*, June 28-July 1 1999.

[69] S. Kaski, and T. Kohonen, “Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World”, *Proc. of the 3rd International Conference on Neural Networks in the Capital Markets*, World Scientific, Singapore, 1995, pp. 498-507.

[70] Fasel, B., Luettin, J.,”Automatic Facial Expression Analysis: A Survey”, *Pattern Recognition 36*, 2003, pp. 259-275

[71] Saeed K. P., Karim F., and F. Hajati, “Face Detection Based on Central Geometrical Moments of Face Components”, *IEEE conference on systems, Man, and Cybernetics*, Oct., 2006, pp. 4225-4230.

[72] Reddy, V.S.K.Sengupta, S. Iatha, and Y.M., **“**A High-Level Pipelined FPGA Based DCT For Video Coding Applications**”,** *TENCON Conference on Convergent Technologies for Asia-Pacific Region***,** Oct. 2003, Vol. 2, pp. 561- 565.

[73] Bruno Santos Pimentel, Joao Hilario de Avila Valgas Filho, Rodrigo Lacerda Campos, Antonio Otavio Fernandes, Claudionor Jose Nunes Coelho, "A FPGA Implementation of a DCT-Based Digital Electrocardiographic Signal Compression Device", *14th Symposium on Integrated Circuits and Systems Design*, 2001.

[74] Yuk Ying Chung, and N.W. Bergmann, "Video Compression on FPGA-Based Custom Computers", *International Conference on Image Processing (ICIP'97)*, Volume 1, p. 361.

[75] Isa Servan Uzun and Abbes Amira, "A Framework for FPGA Based Discrete Biorthogonal Wavelet Transforms Implementation", *13th European Signal Processing Conference*, 2005.

[76] Ali M. Al-Haj, "An FPGA-Based Parallel Distributed Arithmetic Implementation of the 1-D Discrete Wavelet Transform", *Informatica 29*, 2005, pp. 241–247.

[77] A. Lehmann, J. P. Robelly and G. Fettweis, "Design And Automatic Code Generation Of A Two-Dimensional Fast Cosine Transform For SIMD DSP Architectures", *13th European Signal Processing Conference*, 2005.

[78] Hori Y. Shimizu K. Nakamura Y., and Kuroda T**.,** **"**A Real-Time Multi Face Detection Technique Using Positive-Negative Lines-Of-Face Template", *Proc. of the 17th International Conference on Pattern Recognition ICPR*, 2004, pp. 765- 768.

[79] Melanie Po-Leen Ooi, "Hardware Implementation for Face Detection on Xilinx Virtex-II FPGA using the Reversible Component Transformation Color Space", *Proc. of the Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA'06)*, 2006.

[80] Duy Nguyen, David Halupka, Parham Aarabi, Ali Sheikholeslami, "Real-Time Face Detection and Lip Feature Extraction Using Field-Programmable Gate Arrays", *IEEE Transactions On Systems, Man, And Cybernetics—Part B:Cybernetics*, Vol. 36, No. 4, August 2006, pp. 902-912.

[81] A. Georghiades, Yale Face Database, Centre for Computational Vision and Control at Yale University, Available: http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[82] Tirath Ramdas, Li-minn Ang, and Greg Egan, **"**FPGA Implementation Of An Integer MIPS Processor In Handel-C And Its Application To Human Face Detection", *IEEE Region 10th Conference on TENCON*, 2004, Vol. A, pp. 36 – 39.

[81] K. Sandeep, and A. N. Rajagopalan, "Human Face Detection in Cluttered Color Images Using Skin Color and Edge Information", *Proc. of Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2002.

[82] MIT CBCL Face Database #1, MIT Center for Biological and Computing Learning, Available: http://cbcl.mit.edu/software-datasets/FaceData2.html.

[83] P. Viola and M. Jones, "Rapid Object Detection Using A Boosted Cascade Of Simple Features", *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[84] Stan Z. Li and {ZhenQiu} Zhang, "FloatBoost Learning and Statistical Face Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, Vol. 26, Num. 9.

[85] C. Lin, "Face Detection By Color And Multilayer Feedforward Neural Network", *Proc. of IEEE International Conference on Information Acquisition*, June 2005, pp. 518-523.

[86] UMIST Face Database, Vision and Image Processing group, The University of Manchester, Available: http://images.ee.umist.ac.uk/danny/database.html

[87] The ORL Database of Faces, AT&T laboratories, Cambridge University, Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[88] Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, Hong Jiang Zhang, and Harry Shum, "Statistical Learning of Multi-View Face Detection**",** *Proc. of the 7th European Conference on Computer Vision-Part IV*, 2002, pp. 67-81.

[89] Chengjun Liu, "A Bayesian Discriminating Features Method for Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003, Vol. 25, no. 6, pp. 725-740.

[90] Sami Romdhani, Philip Torr, Bernhard Sch¨olkopf, and Andrew Blake, "Computationally Efficient Face Detection", *Proc. of the 8th International Conference on Computer Vision*, 2001.

[91] Paul Viola, and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[92] Paul Viola, and M. J. Jones, "Fast Multi-view Face Detection", *Technical Report, Mitsubishi Electric Research Laboratories*, August 2003.

[93] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja, "Face Detection Using Multimodal Density Models", *the 84th Computer Vision and Image Understanding*, 2001, pp. 264 – 284.

[94] Byeong Hwan Jeon, Kyoung Mu Lee, and Sang Uk Lee, "Face Detection Using a First-Order RCE Classifier", *International Journal on Advances in Signal Processing EURASIP*, 2003, Issue 9, pp. 878-889.

[95] Christopher A. Waring and Xiuwen Liu, "Face Detection Using Spectral Histograms and SVMs", *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, Vol. 35, No. 3, June 2005, pp. 467-476.

[96] M. Yang, N. Roth, and N. Ahuja, "A SNoW-Based Face Detector", *Neural Inf. Process. Syst.*, vol. 12, 2000,pp. 855–861.

[97] H. Schneiderman and T. Kanade, "Probablistic Modeling of Local Appearance and Spatial Relationships for Object Recognition", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 45-51.

[98] Ming-Hsuan Yang, Narendra Ahuja, and, David Kriegman, "Mixtures of Linear Subspaces for Face Detection", *Proc. International Conf. Automatic Face and Gesture Recognition*, 2000, pp. 70-76.

[99] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas," Self-organizing map in Matlab: the SOM Toolbox", *Proc. of Matlab DSP Conference*, Espoo, Finland, November 1999, pp. 30-40.

[100] D. Brown, I. Craw, and J. Lewthwaite, "A SOM Based Approach To Skin Detection With Application In Real Time Systems", *Proc. of the British Machine Vision Conference*, 2001.

[101] P.M. de Zeeuw, "A toolbox for the lifting scheme on quincunx grids (LISQ)", *Report PNA-R0224, Centrum voor Wiskunde en Infprmatica CWI*, December 31, 2002.

[102] Rob McCready,"Real-Time Face Detection on a Configurable Hardware Platform", *A Master Thesis, Electrical and Computer Engineering Department,University of Toronto*, 2000.

[103] Theocharis Theocharides, "Embedded Hardware Face Detection For Digital Surveillance Systems", *A PhD dissertation in Computer Science and Engineering, Pennsylvania State University*, May 2006.
[104] VDEC1 Video Decoder Board http://www.digilentinc.com/Products/Detail.cfm?Prod=VDEC1&Nav1=Products&Nav2=Accessory.
[105] Jonathan B. Ballagh, "An FPGA-Based Run-Time Reconfigurable 2-D Discrete Wavelet Transform Core", *A Master Thesis of Science in Electrical Engineering, Virginia Tech*, June 2001.
[106] Sarma Nedunuri, John Y Cheung, and Prakasa Nedunuri, "Design of Low Memory Usage Discrete Wavelet Transform on FPGA Using Novel Diagonal Scan", *PARELEC Proceedings of the international symposium on Parallel Computing in Electrical Engineering*, 2006, pp. 192-197.
[107] Paschalakis, S. Zakerolhosseini, and A. Lee P., **"**Feature Extraction Algorithms Using FPGA Technology", *High Performance Architectures for Real-Time Image Processing***,** Feb 1998, pp. 1-16**.**
[108] Kotoulas L., and Andreadis I., "Real-Time Computation Of Zernike Moments", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.15, Iss.6, June 2005, Pages: 801- 809
[109] Cristina Cabani & W. James MacLean, "A Proposed Pipelined-Architecture for FPGA-Based Affine-Invariant Feature Detectors", *Conference on Computer Vision and Pattern Recognition Workshop*, 2006, p. 121.
[110] Hung D.L., Cheng H.D., and Sengkhamyong S., "A Reconfigurable Hardware Accelerator For Moment Computation**",** *Proc. of Tenth Annual IEEE International ASIC Conference and Exhibit*, 1997, pp. 238 – 241
[111] Li Xue, Zhao Rongchun, and Wang Qing, "FPGA Based Sobel Algorithm As Vehicle Edge Detector In VCAS", *Proceedings of the International Conference on Neural Networks and Signal Processing*, 2003, Vol.2, pp. 1139 – 1142.
[112] Daggu Venkateshwar Rao, and Muthukumar Venkatesan, "An Efficient Reconfigurable Architecture and Implementation of Edge Detection Algorithm Using Handle-C**",** *Proc. of International Conference on Information Technology: Coding and Computing*, 2004, pp. 843 – 847
[113] Hong Shan Neoh, and Asher Hazanchuk, "Adaptive Edge Detection for Real-Time Video Processing using FPGAs", *Application notes, Altera Corporation*, 2005, www.alter.com.
[114] Ben Khalifa K., Girau, B., Alexandre F., and Bedoui, M.H., "Parallel FPGA Implementation Of Self-Organizing Maps", *Proc. of the 16th International Conference on Microelectronics*, 2004, pp. 709 - 712
[115] Hiroomi Hikawa, "FPGA Implementation Of Self Organizing Map With Digital Phase Locked Loops", *Proc. of the International Joint Conference on Neural Networks*, 2005, Vol.18, Issues 5-6, pp. 514-522.
[116] Georg Pölzlbauer, "Survey and Comparison of Quality Measures for Self-Organizing Maps", *Proceedings of the Fifth Workshop on Data Analysis*, June 2004, pp. 67-82.
[117] Xilinx Online, http://www.xilinx.com/
[118] Simulink Online, http://www.mathworks.com/products/simulink/
[119] System Generator for DSP Online, http://www.xilinx.com/ise/optional_prod/system_generator.htm

[120] Xilinx ISE Foundation Online, http://www.xilinx.com/ise/logic_design_prod/foundation.htm
[121] Xilinx Platform Studio Online, http://www.xilinx.com/ise/embedded_design_prod/platform_studio.htm

Vita

Abdallah S. Abdallah was born in Alexandria, Egypt. He earned his B.Sc degree in Computer Engineering from the Arab Academy for Science and Technology (AAST), Alexandria, Egypt, in August 2003. He joined the military service on January 2003. Then he started working as teaching assistant for computer engineering department, AAST since March 2005 till now. He joined Virginia Polytechnic Institute and State University in fall, 2005, as a direct PhD student in the Virginia Tech Middle East and North Africa (VT-MENA) for graduate studies. His research interests are computer vision, digital image processing, autonomous robots, digital signal processing as well as embedded systems.