SEPTEMBER 2023 M.Sc. in Electrical Electronics Engineering RUSUL NASEER ALLAMI

REPUBLIC OF TÜRKİYE

GAZİANTEP UNIVERSITY

GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES

FACE DETECTION AND RECOGNITION BASED ON RASPBERRY PI USING HAAR CASCADING AND CONVOLUTION NEURAL NETWORK

M.Sc. THESIS

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

BY

RUSUL NASEER ALLAMI

SEPTEMBER 2023

FACE DETECTION AND RECOGNITION BASED ON RASPBERRY PI USING HAAR CASCADING AND CONVOLUTION NEURAL NETWORK

M.Sc. Thesis

in

Electrical Electronics Engineering

Gaziantep University

Supervisor

Prof. Dr. Ergun ERÇELEBİ

by

Rusul Naseer ALLAMI

September 2023

FACE DETECTION AND RECOGNITION BASED ON RASPBERRY PI USING HAAR CASCADING AND CONVOLUTION NEURAL NETWORK

submitted by **Rusul Naseer ALLAMI** in partial fulfillment of the requirements for the degree of Master of Science in **Electrical Electronics Engineering Gaziantep University** is approved by,

Prof. Dr. Çiğdem AYKAÇ
Director of the Graduate School of Natural and Applied Sciences ………............

Prof. Dr. Ergun ERÇELEBİ
Head of the Department of Electrical and Electronics Engineering ………............

Prof. Dr. Ergun ERÇELEBİ
Supervisor, Electrical and Electronics Engineering ………………..
Gaziantep University

Exam Date: 18 September 2023

Examining Committee Members:

Prof. Dr. Ergun ERÇELEBİ
Gaziantep University ………............

Asst. Prof. Dr. Serkan ÖZBAY
Gaziantep University ………............

Asst. Prof. Dr. Erkan KİYMİK
Hasan Kalyoncu University ………............

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

Rusul Naseer ALLAMI

ABSTRACT

Face Detection and Recognition Based On Raspberry Pi Using HAAR Cascading and Convolution Neural Network

Rusul Naseer ALLAMI

M.Sc. in Electrical Electronics Engineering

Supervisor Prof. Dr. Ergun ERÇELEBİ

September 2023

73 pages

Face Detection is a form of biometric method that relates to the automatic detection of faces by computerized systems through observation of the face. It is a popular feature in biometrics, digital cameras, and social tagging. Face detection and recognition have received increased research focus in recent years. In this thesis, a face detection and recognition system have been proposed and developed for detecting and recognition faces through the hybridization of two algorithms: HAAR cascading algorithm and deep learning algorithm. The proposed system consists of two approaches. The first approach, the HAAR cascading algorithm, was developed by taking a shot of the face and reducing it several times to ensure that there is a face at each shrinking time. The second approach has proposed convolution neural network (CNN) model to increase accuracy of classification. In addition to improving each algorithm, hybridization of the two algorithms significantly improved the results of the classification. In proposed system two dataset was used: download dataset, and real dataset. The accuracy of modifying HAAR in detection reached 98.667% for real dataset, and 97.532 % for download dataset. The accuracy of proposed model of CNN in classification reached 96.23% for download dataset, and 100% for real dataset. The tests conducted on the developed facial recognition system have demonstrated that the proposed algorithms and the developed real-time facial recognition system yield satisfactory results.

Key Words: Face Detection, Face Recognition, HAAR Cascading, CNN.

ÖZET

HAAR Basamaklı ve Evrişim Sinir Ağı Kullanılarak Raspberry Pi Tabanlı Yüz Tespiti ve Tanıma

Rusul Naser ALLAMI

Yüksek Lisans Elektrik Elektronik Mühendisliğinde

Danışman Prof. Dr. Ergun ERÇELEBİ

Eylül 2023

73 sayfa

Yüz tespiti, yüzün otomatik olarak bilgisayar sistemleri tarafından algılanmasıyla ilgili bir biyometrik yöntemdir. Biyometrik sistemlerde, dijital kameralarda ve sosyal etiketlemede yaygın bir özelliktir. Son yıllarda yüz tespiti ve tanıma alanında artan bir araştırma odaklanması görmektedir. Bu tezde, yüz tespiti ve tanıma sistemi, iki algoritmanın birleştirilmesiyle (HAAR kademeli algoritma ve derin öğrenme algoritması) yüzleri tespit etmek ve tanımak için önerilmiş ve geliştirilmiştir. Önerilen sistem iki yaklaşımdan oluşmaktadır. "İlk yaklaşım olan HAAR kademeli algoritma, yüzün bir fotoğrafını çekerek bunu birkaç kez küçülterek her küçültme adımında yüzün varlığını sağlamak üzere geliştirildi. İkinci yaklaşım, doğruluk oranını artırmak için bir konvolüsyon sinir ağı (CNN) modeli önermiştir. Her bir algoritmayı geliştirmenin yanı sıra, iki algoritmanın birleştirilmesi sınıflandırma sonuçlarını önemli ölçüde iyileştirdi. Önerilen sistemde iki veri seti kullanıldı: indirilebilir veri seti ve gerçek veri seti. HAAR algoritmasının tespitinde yapılan değişikliklerin doğruluk oranı gerçek veri seti için %98.667, indirilebilir veri seti için ise %97.532 olarak gerçekleşti. CNN modelinin sınıflandırmadaki önerilen doğruluk oranı indirilebilir veri seti için %96.23, gerçek veri seti için ise %100 olarak elde edildi. Geliştirilen yüz tanıma sistemi üzerinde yapılan testler, önerilen algoritmaların ve geliştirilen gerçek zamanlı yüz tanıma sisteminin tatmin edici sonuçlar verdiğini göstermiştir.

Anahtar Kelimeler: Yüz algılama, yüz tanıma, Haar basamaklı, CNN.

''Dedicated to my family''

ACKNOWLEDGEMENTS

Studying for a master’s degree at Gaziantep University is a significant milestone in my life. I want to take this opportunity to thank everybody who supported me on this journey and helped me to finish it better than my pre-expectations.

Many thanks to all my teachers, especially my teacher and supervisor, Prof. Dr. Ergun ERÇELEBİ, for his support and guidance.

My family deserves endless gratitude. My father and my mother patience and my brothers support were the main factors for my success. Big thanks to the best Son in the world Mahdi, I hope you success in your life.

Finally, I will never forget the support and encouragement of my friends and everyone who supported me with prayers that I could not succeed without them.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page*

LIST OFABBREVIATIONS

CCTV	Closed-circuit television
CNN	Convolutional Neural Network
CPU	Control Processing Unit
ERM	Empirical Risk Minimization
GPIO	General Purpose Input Output
HOG	Histogram Orientation Gradient
PIR	Passive infrared sensors
PTZ	pan–tilt–zoom
ReLu	Rectified Linear Unit
SRM	Structural Risk Minimization
SVM	Support Vector Machine

CHAPTER 1

GENERAL INTRODUCTION

1.1 Introduction

The human face is a vital component of daily social interactions because it conveys information about who we are as individuals. Face recognition is a form of biometrics that can be used to positively identify a person. This approach employs mathematics to recognize a person's face in order to determine their characteristics, which are then saved as a digital face print. A growing number of institutions and organizations have recently expressed an interest in its wide range of applications in civilian domains as well as other law enforcement [1]. Unlike fingerprint, palm print, and iris recognition, face recognition does not require any physical contact. Facial recognition software can also identify someone from a distance without needing to make physical contact with them. Crime prevention is further enhanced by using face recognition technology to store images of criminal suspects and use them in a variety of ways, such as to identify a person. At venues like train stations, airports, and bus stops, face recognition software is currently being employed [2]. It is also being used in health care and advertisement. Face recognition necessitates large data sets and sophisticated features in order to uniquely identify individuals in the event of changing stance, illumination, and age. [3]. In comparison to where it was a decade ago, face recognition has advanced significantly. Even when there are only a few persons in the frame, facial recognition technologies typically perform well. In addition, these approaches have been evaluated in well-lit environments with good face expressions and images of high quality [4].

The field of computer vision has made significant progress in the area of face recognition. Face detection and efficient recognition are more difficult than ever because of technological advancements and the prevalence of multimedia applications on smartphones. Since the pioneering work of Viola Jones in detecting the frontal face in real time with low computational complexity, the field of face recognition has developed considerably [5].

Afterwards, after training and recognizing multiple classifiers, numerous methods emerged that made use of fundamental image processing techniques to extract distinct facial features from the provided images. In addition, most of the earliest methods in face recognition relied on upright images with limited variations in illumination, pose, occlusions, etc. Separated face recognition includes a technique of feature extraction and contrast identification of normalized face images to determine the identify of human faces in the images. Because of this, facial recognition methods are crucial for effective surveillance [6]. In order to verify an individual's identification, the Deep Convolutional Neural Network (CNN) method is utilized to compare a live capture or digital image to the stored face print. CNN can effectively communicate the qualities of the chain of importance and the characteristics of the image. Classification of protests, image division, and video recovery are only some of the many uses for the current convolution neural system [7]. The convolution neural system is most useful when applied to learning at a profound level. It can make use of a large amount of labeled data, extract attributes from various levelled structure, and get the depth of the image qualities, allowing for the communication of more nuanced picture aspects [8].

1.2 Related Work

Many previous related works are selected:

a. In 2017, Muhammad Sajjad, et al [9], they've been laid bare. They've been shown. A Raspberry Pi and a cloud-based facial recognition software were suggested. A wireless, small, portable camera was put on a police officer's uniform to record streaming video. These streams were then transferred to a Raspberry Pi for facial detection and recognition. The recommended method uses Bag of Words to extract FAST and BRIEF points from the recognized face, followed by an SVM to identify suspects. Experiment results support the suggested method's precision in recognizing faces and its use for boosting law enforcement services in smart communities.

b. In 2019, Muhammad Zeeshan Khan, et al [10]. In their study, they suggested a system for identifying and recognizing faces based on Convolution Neural Networks

(CNN) that works better than other methods. Face recognition in a smart classroom could be used to test how well the suggested algorithm works. There are no known problems with the way the system performs when the LFW dataset is used to train it. After 35 faces, the system can distinguish 30 of 40 kids apart from a single image. Based on the test results, it was found that the proposed method was 97 % accurate.

c. In 2020, Ahamed Yasar Z, Dinesh Kumar Rand else [11], a border monitoring system is explained in this paper. This is accomplished via computer vision. All that is required is a Raspberry Pi, a camera, a Buzzer, and an Arduino Uno. A digital camera is used to record the live motions. OpenCV (computer vision) software uses the collected data/records as an input on the raspberry pi to detect positive and negative objects and faces. The relevant code is written in Python. When the pi module identifies a face person or an object that's not quite right, the pi will immediately transmit the signal. Central authorities or monitoring rooms can receive whatever signal they want.

d. In 2021, Ratnesh Kumar Shukla and Arvind Kumar Tiwari [12], this article highlights a device that focuses on images and allows the user to recognize and detect a variety of facial features utilizing a webcam. In this paper, they conduct comprehensive and systemic research to evaluate the effectiveness of various traditional representation learning structures on class-imbalanced results. Also demonstrates that deeper discrimination could be taught by creating a network that preserves inter-cluster differences as well as differences within groupings. The recently proposed (CNN) model, MobileNet, delivers both offline and real-time precision and speed to achieve rapid and consistent stable outcomes. One of the most common issues in face recognition and identification has been overcome by this. In order to deal with the problem of faces, a number of researchers have employed various methodologies and models across the literature. In order to achieve the best results, they employ the most layers possible. A machine learning approach combined with multiple image-based datasets improves the classifier's ability to predict knowledge relevant to face detection and recognition.

e. In 2022, Yuhang Lu, and Touradj Ebrahimi [13] Deep convolutional neural networks have done very well at many different tasks involving detecting and recognizing. Nevertheless, the performance of such detectors is frequently tested in public benchmarks under limited and unreal conditions. Noise, compression, and enhancement, which are common distortions and processing processes in image workflows, are not well explored. Few studies have been conducted to enhance the detector's robustness to unanticipated perturbations. This research provides a more efficient data augmentation approach based on the actual process of image degradation. This innovative technique is used for deep fake detection tasks and has been assessed using a more realistic assessment framework. Extensive tests show that the suggested data augmentation method makes it easier to generalize to data distortions and datasets that were not expected.

1.3 Problem Statement

Face detection, face preprocessing, and face recognition are the three stages of a complete face recognition system. To begin extracting the face difference features, the face region must first be extracted from the face detection procedure and separated from the background pattern. Face detection methods that rely on deep learning have recently become popular since they are faster and more accurate than the previous methods. Separated face recognition includes a technique of feature extraction and contrast identification of normalized face images to determine the identify of human faces in the images.

1.4 Aim of thesis

The aim of this thesis is to develop an intelligent monitoring system based on Raspberry Pi hardware. This involves developing face detection and face classification algorithms. summarizing the study conducted in the thesis briefly are:

- To design and implement a system for monitoring.
- To optimize face recognition using convolution neural network (CNN).
- To design hardware monitoring device using raspberry pi4.
- To analyze the performance of the system.

1.5 Content of Chapters

Chapter One: this chapter presents introduction about monitoring system, related work, and aim of thesis.

Chapter Two: The theoretical background of convolutional neural networks, face recognition, and hardware sensors are presented in this chapter.

Chapter Three: The proposed flowchart is included in this chapter.

Chapter Four: The experimental results of the suggested system are presented in this chapter.

Chapter Five: this chapter contains conclusion and recommendation.

CHAPTER 2

Theoretical Background

2.1 Introduction

The need for video surveillance systems has been increasing recently. Closed-circuit television (CCTV) cameras and other sensor systems are increasingly being installed by governments and businesses to ensure the safety of their citizens and consumers [14]. Video surveillance, which is widely used in the customs, military, police, airports, firefighting, railways, urban transport, and a great many other public places, can be regarded as an integrated system with powerful preventative abilities. Due to the fact that it is precise, quick, and contains a lot of data, video surveillance has become an integral aspect of the security system. [15-16]. There are many researchers who have shown a keen interest in the development of intelligent systems of surveillance. This kind of system keeps track of both permanent and temporary objects in a certain environment in real time.

The main goals of these systems are to interpret and understand scenes automatically, as well as to predict the actions and interactions of the monitored objects based on the data collected by the sensors. In these types of surveillance systems, a lot of sensors work together to process data. Objects' locations can be tracked with visual sensors, while their identification can be verified with identity sensors. Also, visual sensors that have a pan–tilt–zoom camera (PTZ camera) can zoom in on objects so that high-resolution data can be obtained. In an intelligent surveillance system, searching the database for information about a surveillance event is one of the most important tasks.

The management of information and the searching procedure have been regarded as key strategies for use in various fields of study, such as the data Military

[17-18]. The automatic identification of fire and the detection of forgery are just a few examples of the surveillance video processing that rely on motion estimate [19].

This chapter introduces a brief coverage on a way for an intelligent surveillance system utilizing convolution neural network for face recognition and HAAR cascade for face detection by using python programming and from Open-CV library.

2.2 Hardware Components

There are many parts of the hardware that are used in the design of monitoring devices. Which are (Raspberry Pi, cameras, sensors, and etc.).

2.2.1 Raspberry Pi

The device known as the Raspberry Pi computer is an excellent choice for usage in classroom because of its low price and small size. It uses open-source Linux and has a GPIO (General Purpose Input Output), and it's based on the Core CPU. Connector for electrical circuitry that can be used to connect to the robot. Connecting a camera to the robot via the Raspberry Pi provides it the ability to observe its surroundings and send images back to the user. [20]. Figure (2.1) shows a Raspberry Pi B.

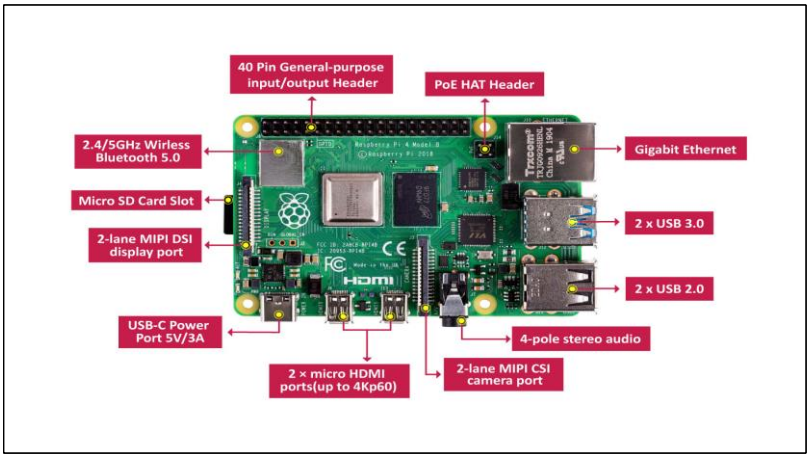


Figure 2.1 Raspberry pi model B [20]

2.2.2 Types of Cameras

Different kinds of cameras are used to connected to raspberry pi, such as USB-cameras, Pi cameras v1 and v2, and intelligent cameras.

Pi camera: Detecting faces through the utilization of the camera module. The Raspberry Pi Foundation is responsible for developing this camera module with 5 megapixels of clearly visible picture quality. Every innovator can easily afford it for camera projects. [21]. As depicted in Figure (2.2).

Figure 2.2 Types of cameras used for raspberry pi 4 [21]

2.2.3 Types of Distance Sensors

a. **Human Detection Unit:** Passive infrared sensors (PIR) which attached to the Raspberry Pi board to detect any human movement. This passive infrared sensor's sensitivity has been set to normal. When the motion sensor detects human movement, the camera connected to the Raspberry Pi board is activated, and capturing of pictures begins. Then, facial recognition software running on the Raspberry Pi board extracts the facial portion from the captured image. [22]. As depicted in Figure (2.3).

Figure 2.3 PIR sensor [22]

b. **Ultrasonic Sensor:** Using this tool, the distance to an object can be determined. Objects between 20mm and 4500mm in length can be disclosed. The device's two digital pins serve as a rough indicator of the measured distance. Up to twenty times per second, a pulse can be activated. [23]. As shown in Figure (2.4).

Figure 2.4 Ultrasonic Sensor [23]

2.2.4 Communication (Router)

A router is employed to facilitate the transmission of data between the sender and the receiver. The primary function of this router is to establish a connection between a Raspberry Pi device and the server-side equipment [24]. Figure (2.5) depicts a router.

Figure 2.5 Wi-Fi adapter [24]

2.3 System of The Face Recognition

The system of the face recognition are useful for many security applications due to their can applied using images, videos, real time recording, independence from the individual (without regard to gender, ethnicity, or hair), robustness in varying lighting conditions, and can work to detect and recognize a face from different angles [25]. As depicted in Figure (2.9) [26]. There are three primary phases used to construct a robust face recognition system like the detection of face, feature extraction, and the last phase is face recognition are. The face detection step is used to detect and locate images of human faces acquired by the system. The vectors of features for faces of human that were determined in Step 1 are extracted in Step 2 of the process known as feature extraction. At the final stage of face recognition, extracted characteristics are compared to a database of template faces in order to make an identity decision [27] as illustrate in Figure (2.6).

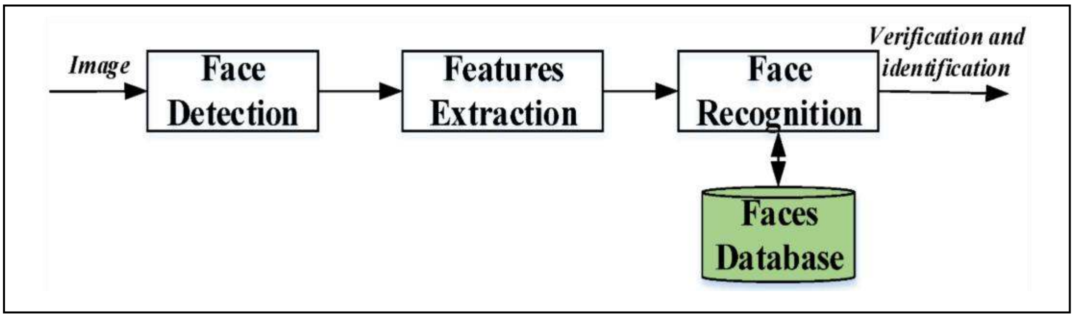


Figure 2.6 Primary phases of face recognition system [27]

2.4 Convolution Neural Network (CNN) based recognition

With an input layer, an output layer, multiple hidden layers, and millions of parameters, a Convolution Neural Network can learn complex patterns and objects. Hidden layers are partially connected, and finally a fully connected layer that leads to an output layer. The input is first subsampled by convolution and pooling, and then an activation function is applied to the resulting data. The resulting image maintains its original shape, in contrast to the input image [28]. Figure (2.12) is an example of this. In what follows, we'll take a look at the CNN components:

A. Convolutional Layer

The convolutional layer is necessary for image classification. The main purpose of this layer is to extract features from the input image. The convolution layer contains multiple feature maps. On the former surface, the same neuronal characteristic map extracts the regional characteristics of many locations. However, for a single neuron, it is extracted via the regional feature map containing the same places as the previous feature map, which the subsequent layer uses as input [29]. Figure (2.7) illustrates a typical convolutional procedure.

The basic building blocks of a convolutional neural network (CNN) are called "convolutions," and they are found in the layers of the network. The following are typical features of this layer:

1- Input of CNN algorithm is vectors (Image)

2-The filters is used as features detector.

3-Vectors of output (Feature map), where input image x

$$Feature\ Detector\ = \sum_{i}^{n}\sum_{j}^{m} P_{i,j} \times Kernal \qquad (2.1)$$

Where n is map width, m is map height, P is pixel value, and kernel is features extraction filter.

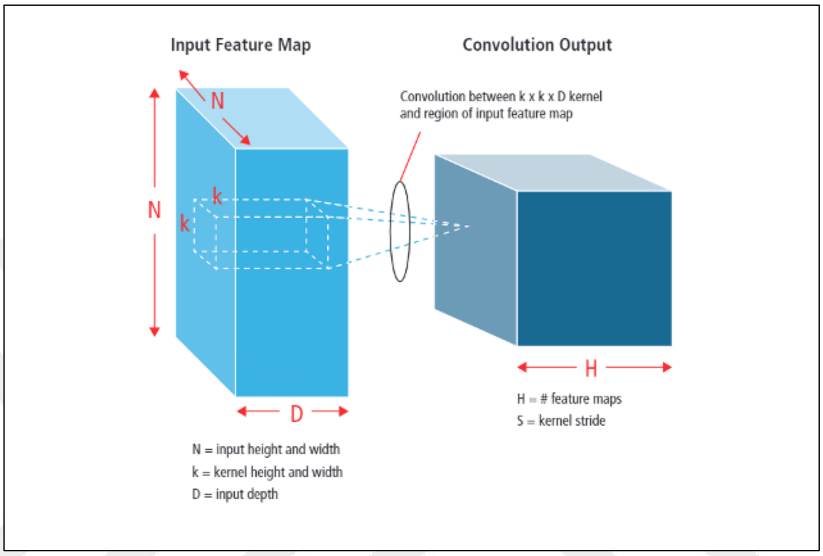


Figure 2.7 convolution layer [29]

B. Rectified Linear Unit (ReLu)

It is a part of the convolutional layer that is responsible for conducting the thresholding operation, which means that a value less than zero is set to zero. As a result, there is less duplication of data, and the most important aspects are unaffected. The output for this layer will be the same size as the output for the previous layer, [30] It is represented mathematically as:

$$maxpooling = Max\left(X_{i,j}\right) \qquad (2.2)$$

.

Where X is input, and (i, j) are indices.

C. Pooling and Subsampling layers

Each feature map is processed independently by the pooling layer. Hence, the feature map's resolution is decreased while the essential elements for classification remain retained. Down-sampling is when you take smaller samples. As depicted in Figure (2.8), there are a number of methods to pool resources [31]:

1.Max-pooling: Max-pooling refers to a method of selecting the maximum elements from the feature map. The max-pooled layer that is the outcome contains essential data from the feature map. This is the most popular strategy because it produces better outcomes [31].

2.Average pooling: This types will calculate the average for each patch on the feature map when calculating the average [32].

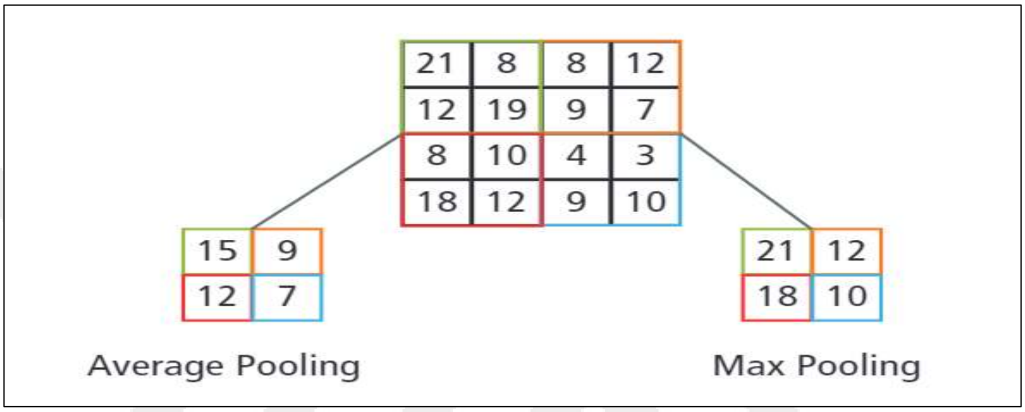


Figure 2.8 Max Pooling [32]

Square data of size 4x4 is provided as input. Subsampling by a factor of 2x2 involves subdividing a 4x4 image into four 2x2 matrices that don't overlap. The output of max pooling is the highest possible value from the four columns of the 2x2 matrix. The result of average pooling is calculated by taking the average of the four input values. It is crucial to consider that the mean value for the output at index (2,2) is a fractional number, which has been rounded to the nearest integer for the purpose of providing information [33].

D. Fully Connected layer

All of the neurons in this layer will have extracted and connected to the neurons in the previous layer (the max pooling layer). The final number of classification classes is the output of this layer. Overfitting can be mitigated by employing the regularization technique dropout in neural networks. It's a technique wherein some of the network's nodes are ignored at random while being trained. Its normal dropout rate is 0.5, and it can be changed for optimal results and faster training [34]. Essential network properties can be learned and generalized as a result of this regularization. illustrate in Figure (2.9).

E. Soft-Max Layer

The activation layer known as soft-max is frequently used as a classifier in the final layer of a network. Classification of input into different classes occurs in this layer. The soft max function provides a mapping between the unnormalized output of a network and a probability distribution. The output of this layer is sent into the soft max layer, where it is transformed into probabilities. Each class in a multi-class problem is given a decimal probability that adds up to 1.0 using soft-max. The result can then be interpreted as a probability. The logistic function is used for two-class problems, while the soft-max method is employed for multiclass problems [35], as illustrate in Figure (2.9).

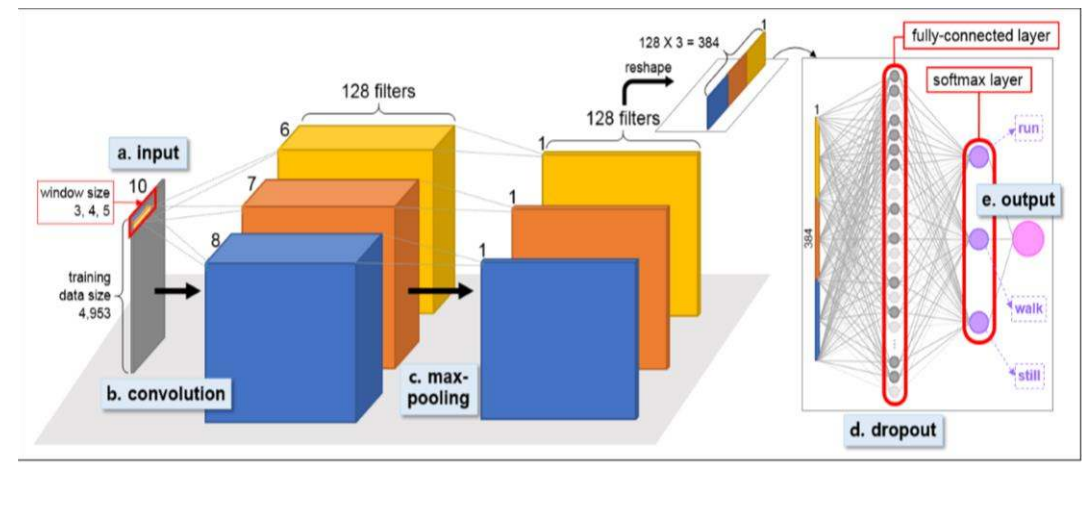


Figure 2.9 General layers of CNN algorithm [34]

2.5 Viola and Jones Algorithm

Paul Viola and Michael Jones have introduced a useful technique for object detection using HAAR feature-based cascade classifiers. Using a method based on adaptive machine learning, a cascade function is trained using both positive and negative images. From there, it can be applied to the task of finding objects in the other images. To get started, the classifier has to be trained on a large number of positive images (people image), as well as negative images (image for objects without people in them). The data is then utilized to generate features, of which there are four primary concepts [36].

A. HAAR-like features: utilized by the HAAR cascade classifier to detect human faces. There are three motions for HAAR-like features. From Figure (2.10), Edge features come first, followed by line features and finally four rectangle features. Fast calculation is guaranteed by the HAAR-like principle. A term for this is "HAAR-like features." [37].

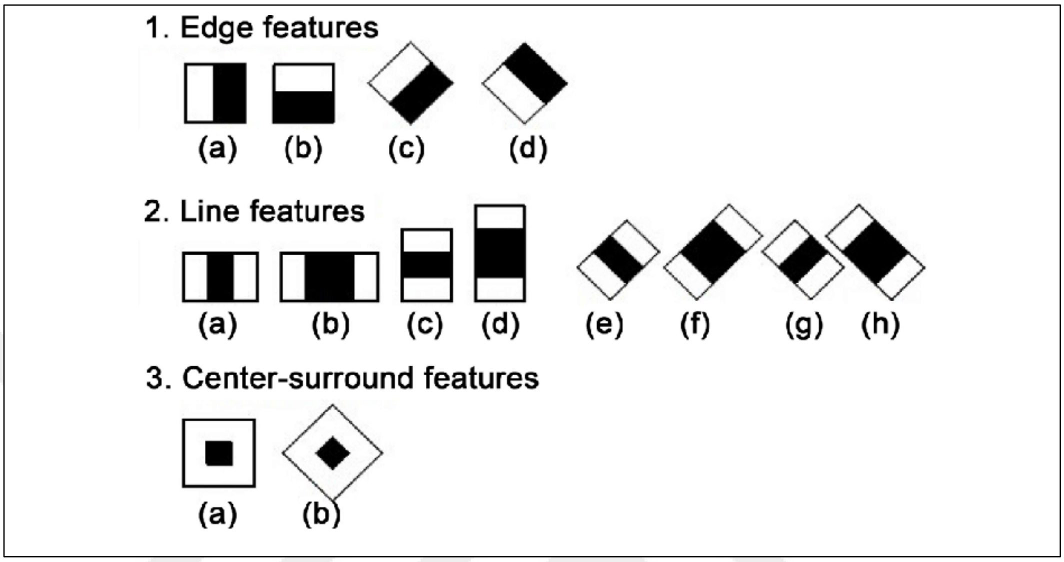


Figure 2.10 HAAR-like features [37]

Subsequently, the features are extracted. Each feature is computed as the difference between the sum of the pixel values below the white rectangle and the sum of the pixel values below the black rectangle. These numerical values are then used to distinguish across images [38].

B. Integral Image: Taking the input image and transforming it into an integral image is the second stage in the Viola-Jones face detection algorithm. Summation of values in a rectangular subset of a pixel grid is quickly and efficiently calculated using an integral image [36], also called a summed region Table, illustrate in Figure (2.11). Pixels above and to the left of $\acute{x}, \acute{y}$, are added to form the integral image at the x, y location, consist of:

$$ii(x, y) = \sum_{\acute{x} \leq x, \acute{y} \leq y} i(\acute{x}, \acute{y}), \qquad (2.3)$$

Where $the\ original\ image's\ pixel\ value\ is\ denoted\ by\ ii(\acute{x}, \acute{y})$

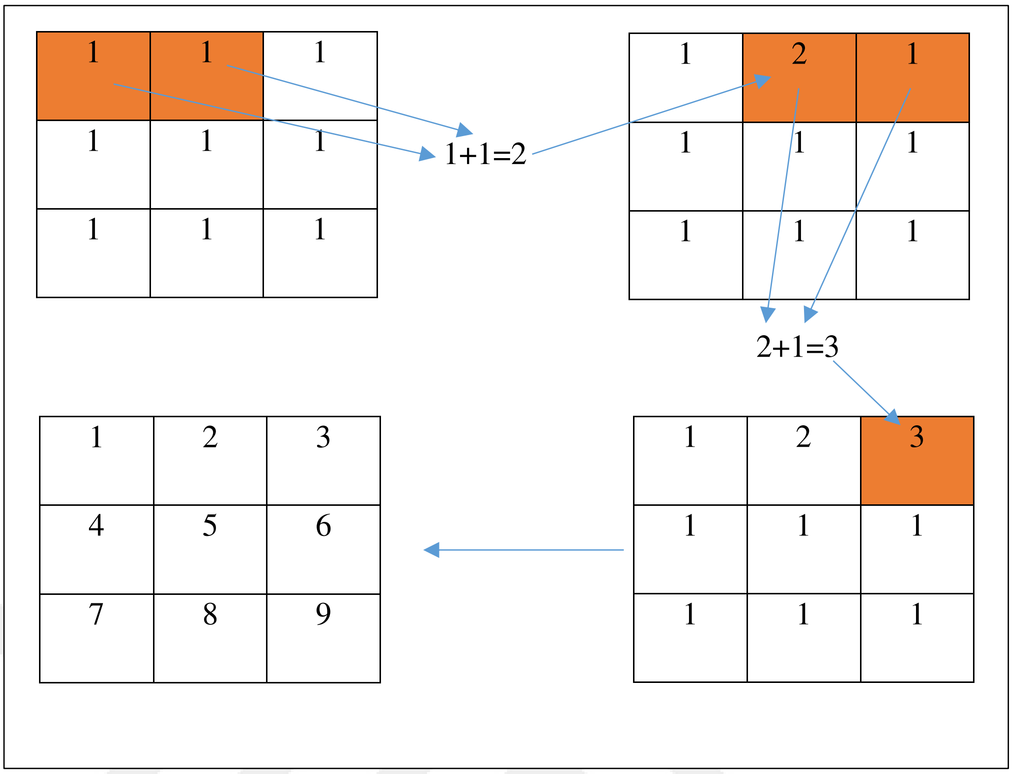


Figure 2.11 Example of an integral image [36].

In the Figure (2.11), and the Integral Image Equation (2.4), demonstrate the ability of employing the integral image to efficiently determine the addition of any rectangular area, where $ii(\acute{x}, \acute{y})$ is the analogical image's integral value. Only four values from the integral image are needed to calculate the addition of the pixels in the rectangle ABCD. Figure (2.12) illustration is the summation of every pixel above and to the left [37].

$$\sum_{(x,y)\in ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C) \ldots\ldots\ldots\ldots\ldots \text{(2.4)}$$

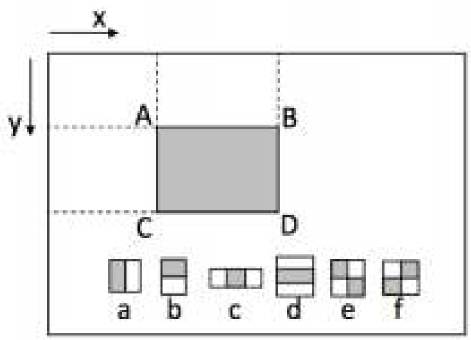


Figure 2.12 Illustration of the integral image [37]

C. AdaBoost Learning Method is a machine-learning method that connects weak classifiers that all have at least one Haar-like feature. This makes a strong classifier. The final classifier is a composite of the weak ones, weighted according to the error they had, thus its accuracy depends on how well each of the weak ones classifies the sample that was applied to it. as shown in the Figure (2.13). The definition of a single weak classifier is [36]:

$$h(x, f, p, \theta) = \begin{cases} 1 & p \times f(x) < T_\theta \\ 0 & otherwise \end{cases} \quad (2.5)$$

Where

x: is a subwindow that has a basic precision of 24×24 pixels.

P: is the polarity

f: indicates the value of the feature

T_θ: is the threshold

A face or a not-face decision will determine the threshold. As can be seen in algorithm (2.6), Viola –Jones adjusted the Adboost algorithm such that it chooses only the more advantageous features.

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t \times h_t(x) \geq L_t \\ 0 & otherwise \end{cases} \quad (2.6)$$

α_t: $\log \frac{1}{\beta_t}$

L_t: is picked in such a way that every positive training sample is classified appropriately.

βt = ∑t / (1- ∑t): the training error of the classifier h_t.

T rounds: meaning construct weak classifiers [37].

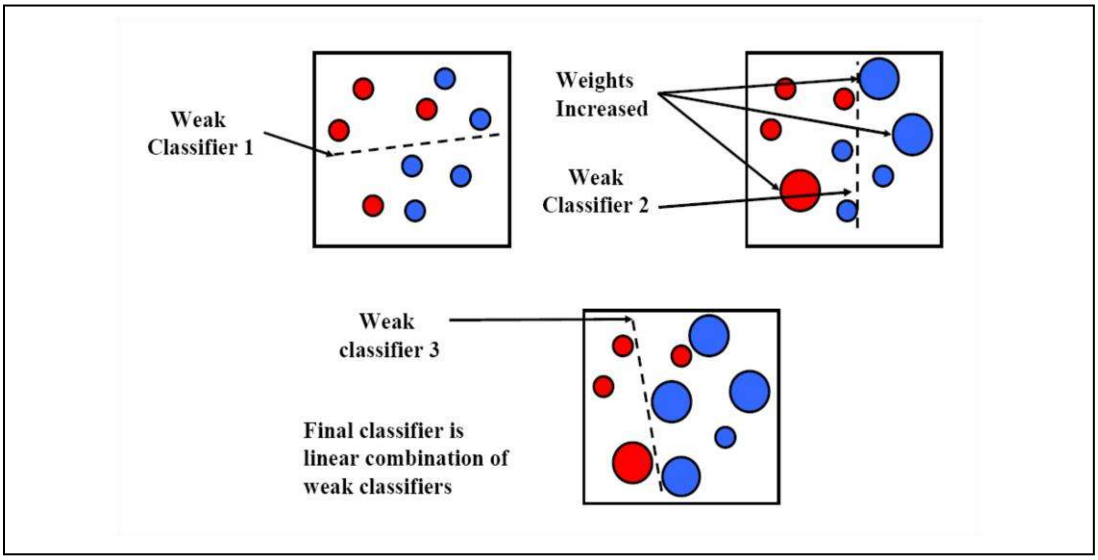


Figure 2.13 strong classifier. [37]

The Adaboost strategy works to determine the optimal polarity and threshold values for a given task. Viola-Jones have devised a proficient and straightforward methodology for identifying each novel weak classifier. This is achieved by estimating various features of all the training instances, with the goal of determining the optimal execution features. The term "weight error" pertains to an error function that is associated with the training samples utilized [38].

D. Cascade Classifiers: Cascade classifier designs are employed to build a series of effective classifiers that can quickly remove unwanted parts of an image [60]. So, the basic classifier puts less emphasis on regions with unlikely to include face and the more advanced classifier prioritizes regions with high likely to include face. Display cascade work as depicted in Figure 2.14. This method drastically reduces calculation time while significantly increasing detection execution. Figure (2.15) is Cascade Classifiers [39].

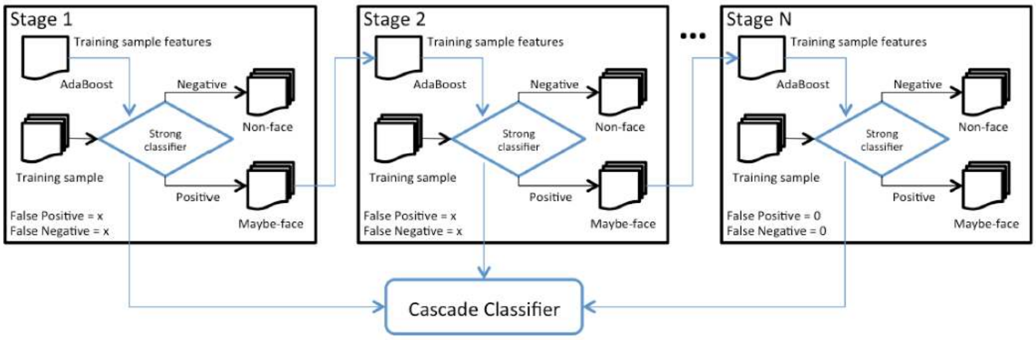


Figure 2.14 Cascade work flow. [39]

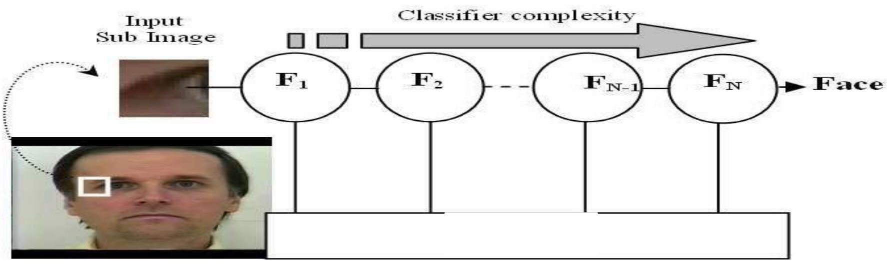


Figure 2.15 Cascade classifier. [38]

2.6 Image Optimizations Using Filters

2.6.1 Mean Filter

This is the simplest form of the mean filter, which is used to eliminate noise from images. There are various kinds of noise for which mean filters are effective. Given a rectangular sub-image with dimensions of n × m, centred at the coordinates (x, y), let S denote the collection of coordinates within this window. The estimated arithmetic average value of the restored picture at each point (x, y) was computed by considering the pixels inside the area denoted by S, and the operation of arithmetic average filtering computes this average value for the distorted image g (x, y). **Figure** (2.16) displays the results of applying a mean filter [48].

10	41	9	3	123
232	186	79	21	134
1	43	56	1	12
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0

Mean
(10,41,9,232,186,79,1,43,56)

10	41	9	3	123
232	186	79	21	134
1	43	56	1	12
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0

Mean
(41,9,3,186,79,21,43,56,1)

10	41	9	3	123
232	186	79	21	134
1	43	56	1	112
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0

Mean
(9,3,123,79,21,134,56,1,112)

$$\frac{10+41+9+232+186+79+1+43+56}{9} = 73.0$$

$$\frac{41+9+3+186+79+21+43+56+1}{9} = 48.7$$

Figure 2.16 Mean filter [48]

The main advantages of using a mean filter include:

A. **Noise Reduction:** Mean filters are effective at reducing noise in an image. They average out random variations in pixel values caused by sensor noise, electronic interference, or other factors. This makes the image appear smoother and less speckled.
B. **Edge Preservation:** Mean filters tend to preserve the edges and important features in an image better than some other smoothing filters, such as Gaussian filters. This is because they have a more uniform kernel that doesn't place as much emphasis on the central pixel [49].
C. **Simplicity:** Mean filters are computationally simple and require minimal memory, making them efficient for real-time or resource-constrained applications. They are easy to implement and execute quickly.
D. **Adjustable Strength:** The degree of smoothing can be controlled by changing the size of the kernel or window. A larger kernel will produce a stronger smoothing effect, while a smaller kernel will result in a milder smoothing effect.
E. **Versatility:** Mean filters can be applied to various types of data, not just images. They are used in signal processing, audio processing, and other fields for smoothing or averaging data points [50].

While mean filters have several advantages in image processing, they also come with certain disadvantages and limitations. Here are some of the disadvantages of mean filters:

A. **Loss of Detail:** Mean filters are primarily used for smoothing and noise reduction. However, their averaging operation can lead to a loss of fine details and subtle textures in an image. This blurring effect may not be suitable for applications where preserving sharp edges and fine features is essential.

B. **Lack of Control:** Mean filters offer limited control over the level of smoothing. You can adjust the degree of smoothing by changing the size of the kernel, but this control is not as fine-grained as some other filters like Gaussian or bilateral filters. It may be challenging to balance noise reduction with the preservation of important image features [51].

C. **Limited Effectiveness with Non-Gaussian Noise:** Mean filters are most effective at reducing Gaussian noise, which has a bell-shaped probability distribution. They may not perform well with other types of noise, such as salt-and-pepper noise or speckle noise. More specialized filters may be required for handling such noise types.

D. **Uniform Averaging:** Mean filters apply a uniform averaging operation to all neighboring pixels within the kernel. This can lead to issues when processing images with non-uniform noise or when there are strong gradients in the image. In such cases, other filters that consider the similarity of neighboring pixels, like the bilateral filter, may be more suitable [52].

In summary, while mean filters are simple and computationally efficient for noise reduction and smoothing in images, they are not always the best choice for every image processing task. The choice of filter should be based on the specific requirements of the application and the characteristics of the image being processed.

2.6.2 Median Filter

While median filters can provide a similar smoothing effect, using a big mask size will result in a more painted (and blurred) appearance. Significant morphological filtering is produced by the median calculation. At first, you'll need to define a neighborhood for each pixel in the image. Each pixel will have its own associated neighborhood, which will likely contain some surrounding source pixels, this might be as simple as creating a square box and centering it on each of the pixels, or it could be more complex, in which case each of the pixels will have its own connected neighborhood, which will comprise some source pixels that are located in the surrounding area. After that comes the step of calculating the median for each of the neighborhoods and putting the results away somewhere secure. In the end, we replaced every pixel with the median value of the neighborhood it was associated with [49]. The effect of applying a median filter is illustrated in Figure (2.17).

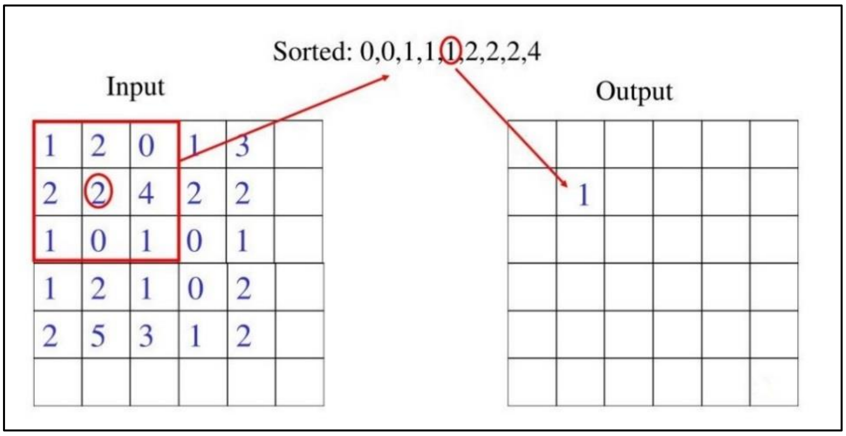


Figure 2.17 median filter [49]

Here are the advantages and disadvantages of using a median filter:

Advantages of Median Filter:

A. **Effective Noise Reduction:** The median filter is particularly effective at reducing various types of noise, including salt-and-pepper noise, impulse noise, and random noise. It can effectively remove isolated noisy pixels without blurring the image.

B. **Preservation of Edges and Details:** Unlike mean filters that tend to blur edges and fine details, the median filter preserves sharp edges and fine structures in an image. It achieves noise reduction without compromising image quality.

C. **Non-Linear Operation:** The median filter's non-linear operation makes it suitable for removing outliers and extreme values in an image. It is robust in the presence of outliers and does not get easily influenced by them.

D. **Adaptive Smoothing:** The degree of smoothing is adaptive and depends on the local pixel values within the kernel. It can effectively adjust to the local image characteristics [52].

E. **Simple Implementation:** The median filter is relatively easy to implement, and its computational complexity is reasonable for real-time applications.

Disadvantages of Median Filter:

A. **Loss of Fine Details:** While the median filter is excellent at preserving edges and major structures, it can still lead to some loss of fine details, especially when strong noise is present. In some cases, this may not be suitable for tasks that require detailed image analysis.

B. **Kernel Size Selection:** The choice of kernel size in the median filter is crucial. A small kernel may not effectively remove noise, while a large kernel may lead to excessive smoothing and the loss of important details. Selecting the appropriate kernel size can be challenging [51].

C. **Computationally Intensive for Large Kernels:** When using large kernels or windows, the computational requirements of the median filter increase significantly. This can be a drawback for processing high-resolution images in real-time applications.

D. **Edge Artifacts:** The median filter can introduce slight artifacts near edges in the form of staircase-like patterns. This effect is more pronounced with larger kernel sizes.

E. **Not Suitable for All Noise Types:** While the median filter is effective for impulse noise and some random noise types, it may not be the best choice for reducing Gaussian noise or other specific noise patterns. Different filters or noise reduction techniques may be more appropriate in those cases [50].

In summary, the median filter is a valuable tool for noise reduction in images, especially when dealing with impulse noise or when preserving edges and details is essential. However, it is not a one-size-fits-all solution, and its effectiveness depends on the specific noise characteristics and image processing requirements. Careful consideration of the filter's parameters and the nature of the image and noise is essential when using the median filter.

2.6.3 Gaussian Filter

As the size of the mask rises, Gaussian produces a more natural effect with less obvious blurring. With a Gaussian filter, the blur is decreased. Gaussian filters give the quickest pulse rise time without overshoot or ringing in the time domain, making them ideal for filtering pulses. The optimal Gaussian filter characteristic is used. A Gaussian blur wherein pixels closer to the kernel's center are given greater weight than those further from the center, as indicated by the filter's corner attenuation frequency and corner attenuation, which are additional parameters. This averaging is performed channel-by-channel and then, it averaging determines the new filtered pixel value [50].

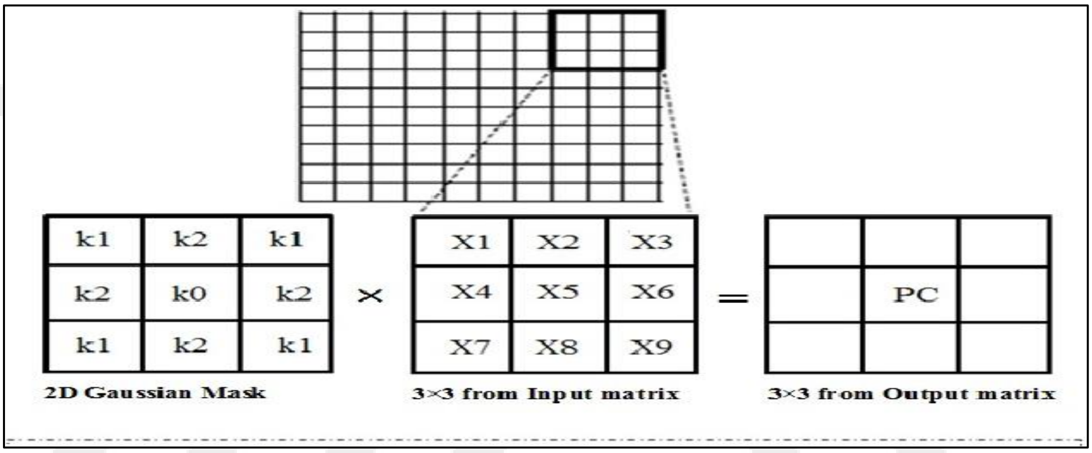


Figure 2.18 Gaussian filter [50]

Here are the advantages and disadvantages of using a Gaussian filter:

Advantages of Gaussian Filter:

A. **Effective Noise Reduction:** Gaussian filters are effective at reducing Gaussian noise and some types of random noise. They achieve noise reduction by smoothing pixel values based on their proximity to neighboring pixels, with more weight given to nearby pixels [49].

B. **Preservation of Image Structure:** Gaussian filters tend to preserve the underlying structure and edges in an image better than simple mean filters. This means that important features and details are less likely to be blurred or distorted.

C. **Adjustable Smoothing:** The degree of smoothing can be controlled by adjusting the standard deviation (σ) of the Gaussian kernel. Smaller values of σ result in mild smoothing, while larger values result in stronger smoothing. This allows for flexibility in achieving the desired level of noise reduction.

D. **Mathematically Well-Defined:** The Gaussian filter is well-defined mathematically, making it a reliable choice for various image processing tasks. Its behavior is smooth and continuous, which can be advantageous in certain applications [51].

E. **Linear and Shift-Invariant:** Gaussian filtering is linear and shift-invariant, which means it can be efficiently implemented using convolution techniques and can be applied to different image regions without modification.

Disadvantages of Gaussian Filter:

A. **Loss of Fine Details:** While Gaussian filters are effective at noise reduction, they can still lead to some loss of fine image details, especially when stronger smoothing (larger σ values) is applied. This may not be suitable for tasks requiring high levels of detail preservation.

B. **Smoothing of Edges:** Gaussian filters do blur edges to some extent. While they preserve edges better than mean filters, there can still be a noticeable reduction in edge sharpness when strong smoothing is applied.

C. **Parameter Sensitivity:** The effectiveness of the Gaussian filter depends on the choice of the standard deviation (σ). Selecting an inappropriate σ value can result in either inadequate noise reduction or excessive blurring. Finding the right σ requires some trial and error [52].

D. **Not Ideal for All Noise Types:** Gaussian filters are optimized for reducing Gaussian noise and may not perform as well with certain non-Gaussian noise types, such as impulse noise or salt-and-pepper noise. In such cases, other filtering techniques like median filtering may be more appropriate.

E. **Computationally Intensive for Large Kernels:** When using large kernels (large σ values), the computational requirements for Gaussian filtering increase significantly. This can be a limitation for real-time processing or high-resolution images [50].

In summary, Gaussian filters are a versatile tool in image processing, providing effective noise reduction while preserving image structure to a reasonable extent. However, they are not without their limitations, particularly regarding the trade-off between noise reduction and detail preservation. The choice of filter and its parameters should be made based on the specific requirements of the image processing task and the characteristics of the image and noise.

Now will review the difference in deleting noise by changing the sequence of use of filters:

Case 1: by applying random choice to filter (Gaussian filter, Mean filter, and median filter) to the original image A we see a difference after applying case 1. illustrated in Figure (2.19)

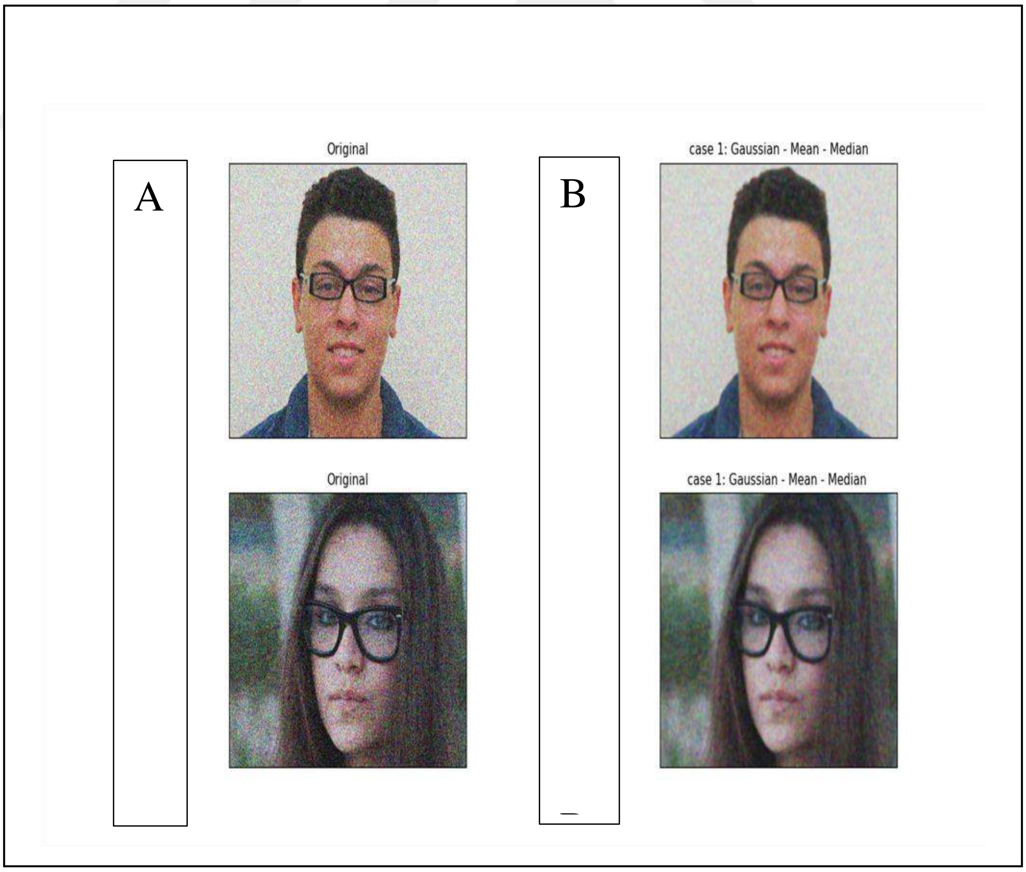


Figure 2.19 Case 1 Gaussian filter, Mean filter, and median filter

Case 2: By applying random choice to filter (median Filter, Gaussian Filter, and Mean Filter) to the original image A, we see a difference after applying Case 2. As illustrated in figure (2.20).

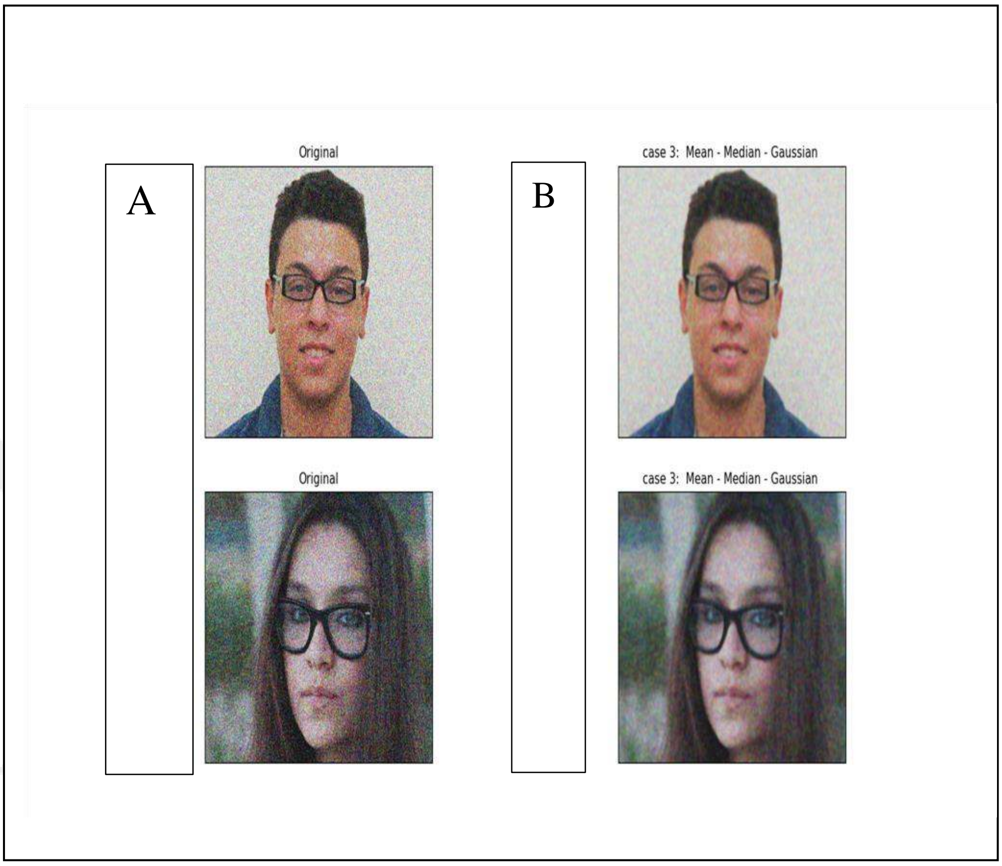


Figure 2.20 Case 2 median Filter, Gaussian Filter, and Mean Filter

Case 3: By applying random choice to filter (Mean Filter, median Filter, Gaussian Filter) to the original image A, we see a difference after applying Case 3. As illustrated in Figure (2.21).

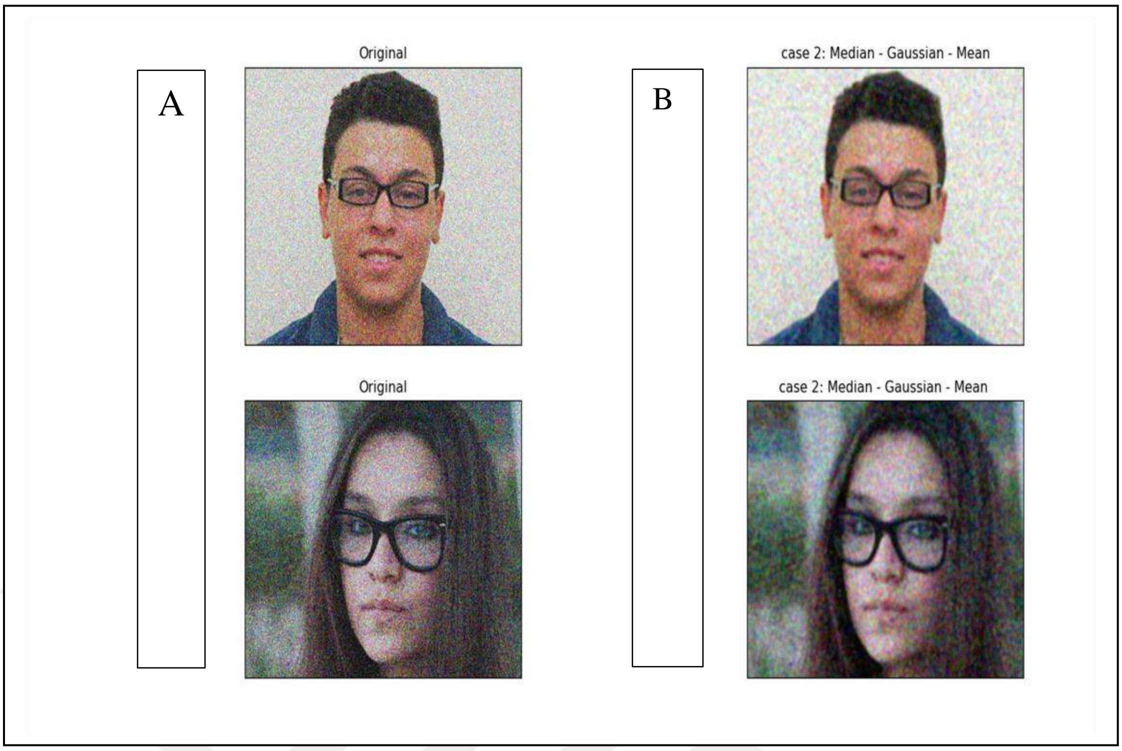


Figure 2.21 case 3 Mean Filter, median Filter, Gaussian Filter

After testing the cases on the original image A, Case 3 was chosen because it showed the best results and used to enhanced image. Figure (2.22) illustrates the best result among the cases.

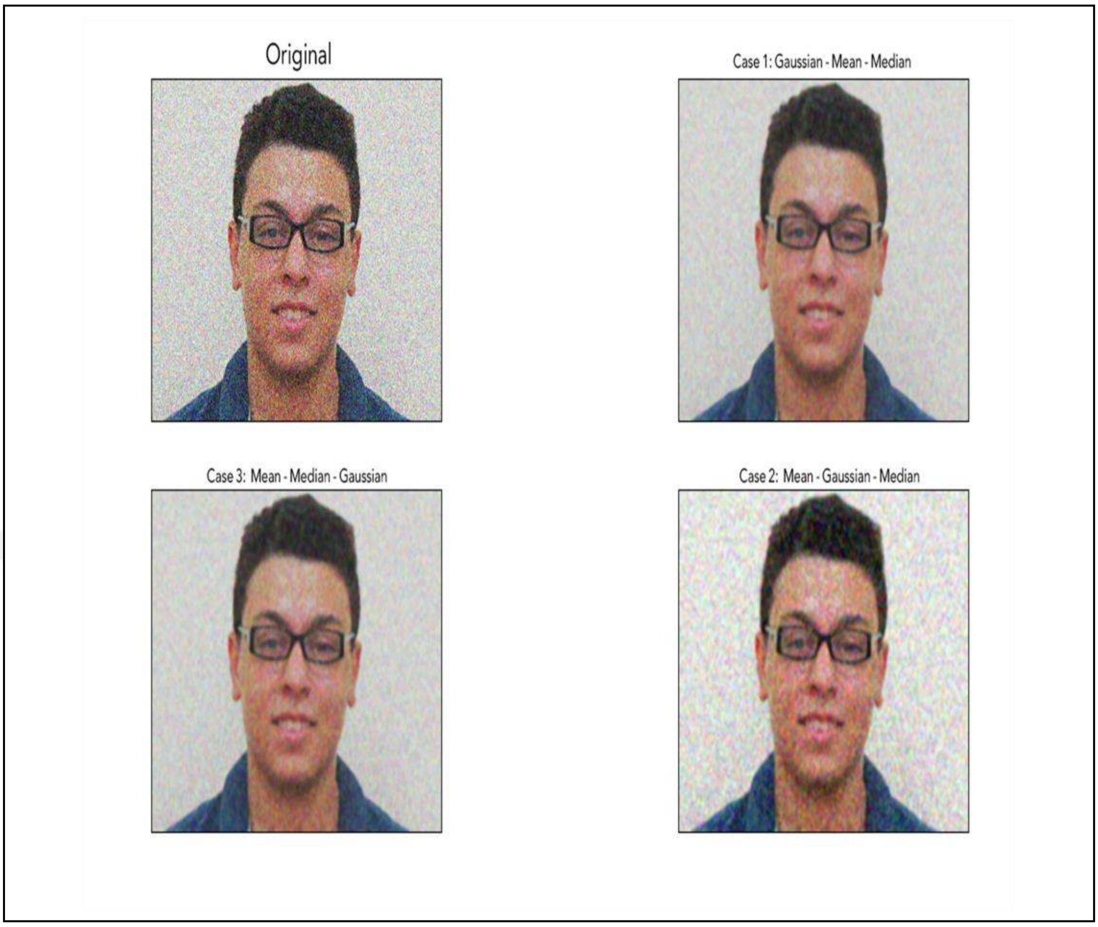


Figure 2.22 The difference between cases

2.7 Database systems

In the field of computer vision, a dataset is an organized set of pictures used to train, test, and evaluate the effectiveness of various algorithms. Using two different face databases, the suggested system and two different models were tested. The first database was obtained over the internet (the Kaggle website), and the other was captured by a mobile camera. The dataset type is front-face. Among the final steps of a surveillance system is data storage and retrieval (valuable features such as user interface design and alarm managing). Through a specific study, a process was carried out to save and retrieve private data in all monitoring data Through a specific study, a process was carried out to save and retrieve private data in all monitoring data.

2.8 Face Recognition Library

Open-CV is an established and widely used open-source computer vision library. It was originally developed by Intel, but it has recently received funding from Willow Garage. Facial recognition, mobile robotics, motion tracking, and artificial neural network are only some of the examples of library applications [66]. As a multi-platform framework, Open-CV offers the advantage of supporting Windows, Linux, and Mac OS X. Open-CV can be intimidating to novices because it offers so many options. To use Open-CV efficiently, you must have a comprehensive understanding of how various approaches work. Thankfully, it is only necessary to possess a few prerequisites.

2.9 Multifarious Aspect in Evaluation Metrics

Recently, numerous different criteria have been employed to gauge the performance of facial recognition systems. Many of these examples are provided in the next section. The application of ground truth principles for negative and positive detection in facial recognition system evaluations [51-52] has been popular for a long time. Table 2-3 depicts the confusion matrix. In any disclosure task, the absence of symmetry is made clear by the use of both positive and negative terms if one class category can be the related class category but the other cannot relate. It is important that the system can distinguish between human and non-human evaluation methods. All the fake videos that were detected as such are called true negatives, whereas all the non-human images that were incorrectly identified as faces are called false negatives. Those who match with images in the database via the system are considered true positives, whereas persons who match with images that are not in the database are considered false positives [53].

Table 2.1 Confusion matrix.

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

There are two significant evaluative plots, namely PR (precision and recall)

1. **Precision**

It's the percentage of detected faces by the system whose likenesses appear in the database. Furthermore, it indicates the level of trustworthiness. The Equation 2-7 exemplifies them [54].

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \qquad \textbf{(2.7)}$$

2. **Recall:** Here, we are tracking the percentage of positive cases. As demonstrated by this study [55], the recall rate of true positives may also be employed to indicate sensitivity. It is computed by utilizing Equation 2-8.

$$\boldsymbol{Recall\ = \frac{TP}{TP+FN}} \qquad \textbf{(2.8)}$$

3. **Recognition rate and error rate**

The term "recognition rate," which refers to "accuracy," is defined as follows [55]:

$$\boldsymbol{Recognition\ rate = \frac{TP}{Total\ number\ of\ test\ images} \times 100\ \%} \qquad \textbf{(2.9)}$$

However, the definition of the error rate is:

$$\boldsymbol{Error\ rate = \frac{FP}{Total\ number\ of\ test\ images} \times 100\ \%} \qquad \textbf{(2.10)}$$

CHAPTER 3

PROPOSED SYSTEM

3.1 Introduction

Face detection and person recognition systems have evolved significantly as they have been used in many areas, but they have suffered from a number of problems and it was necessary to develop a system for face identification and recognition. In this chapter the algorithms and techniques proposed in this thesis will be illustrated.

3.2 Dataset Description

Deep learning necessitates a big quantity of data because it learns features directly from the data using an end-to-end procedure. In this thesis, the following two datasets were utilized:

a. The initial dataset has been obtained from the internet **(Face Recognition Dataset)** [56] and is utilized to test the detection and recognition techniques utilized in the study. The 320 images in this dataset are divided into 40 classes, each of which comprises 8 samples, and these samples show images that include faces. Recognition of faces is enabled by the dataset.

b. For testing the designing device ability to detect face and recognize them, the second dataset is a real archive from a real individual. This dataset has 150 images of faces, each of which is categorized into ten classes, each of which contains 15 samples. With the help of facial recognition, the designing is able to gather information about its surroundings.

Datasets have been divided between training and testing parts in this study, as follows:

a. Training part: 70% of the dataset is represented by these 224 images. The CNN is used to train the system using these images.

b. Testing part: 30% of the dataset can be found in this collection of 96 images. These images are employed to evaluate the precision of the proposed method for classifying and identifying individuals.

Real data has been divided into:

a. Training part: 70% of the dataset can be found in this collection of 105 images.

b. Testing part: 30% of the dataset may be found in this collection of 45 images. The pool layers in the CNN algorithm were used to pull out the features of each image. Which features were extraction depending on the algorithm that was used.

3.3 Proposed System Architecture

The proposed system consists of three main parts, the first of which is the proposed device to face surveillance, the second part is the development of an algorithm to identify faces, and the third part is the development of a model using a deep learning algorithm, the aim of which is to obtain high accuracy in distinguishing people. Figure (3.1) illustrate the general architecture of proposed system.

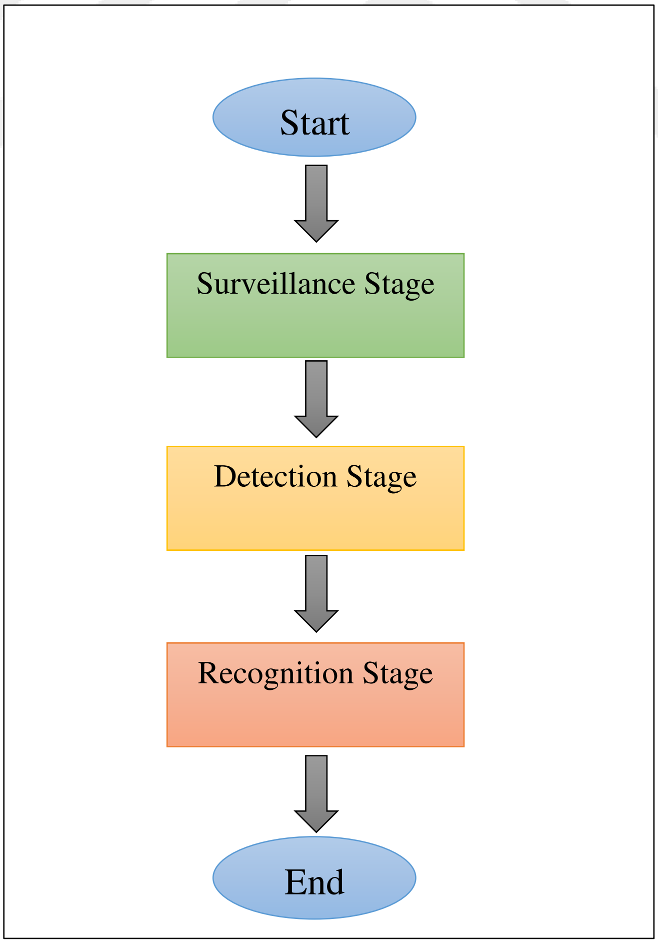
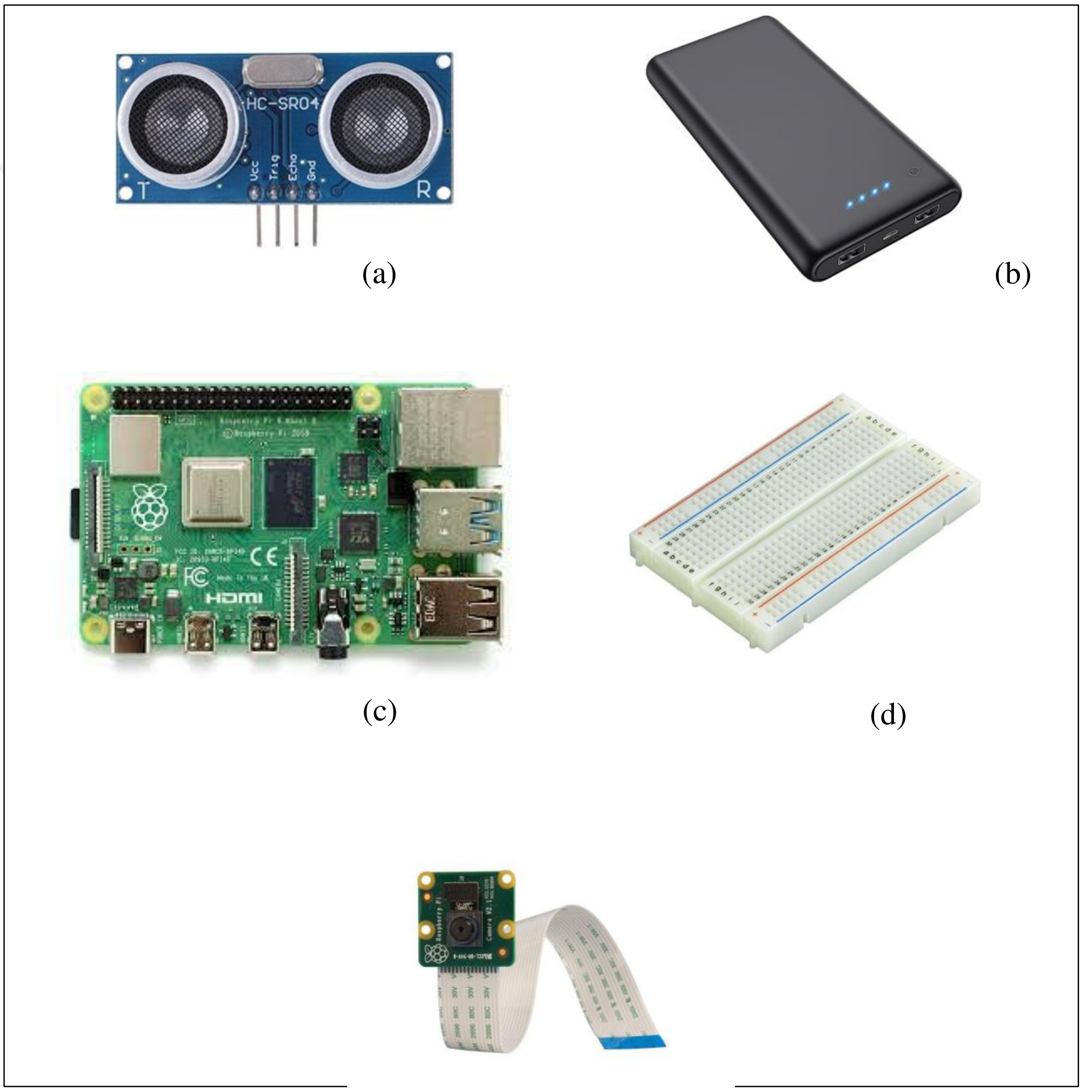


Figure 3.1 Proposed System Stages

3.4 Surveillance Stage

In this section, the proposed surveillance stage will be explained, as a monitoring device has been designed through which people are monitored and their faces detected. The monitor is designed based on the Raspberry Pi 4 controller. The proposed device consists of a set of parts, namely the main controller, the distance sensor, the surveillance camera, in addition to the resistors, breadboard, electrical power sources, and etc. Figure (3.2) illustrate parts of hardware designing device.



(a) (b)

(c) (d)

(e)

Figure 3.2 Hardware Parts (a)Ultrasonic distance sensor (b) Power bank (c) Raspberry Pi4 (d) Breadboard (e) Camera

Where the distance sensor was connected to the main controller, using a number of connecting wires and two resistors, one of which is 1 kΩ and the second 2 kΩ, the reason is to regulate the electrical energy entering the sensor. The use of this sensor is intended for the purpose of determining whether a person is within surveillance or not. Figure (3.3) illustrate electrical circuit of distance sensor. Algorithm (3.1) illustrate distance sensor works.

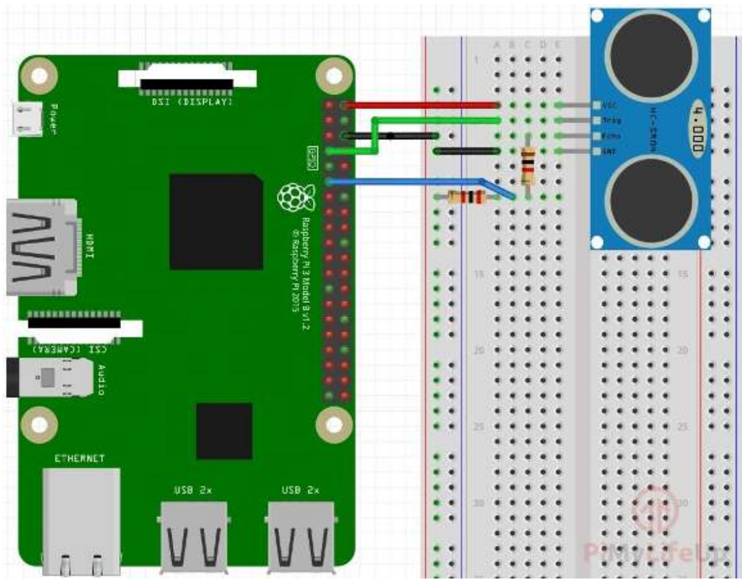


Figure 3.3 Distance sensor circuit

Algorithm (3.1) Distance Sensor Work
Input: Signals
Output: Distance
Begin: Step1: Define two variables (Trigger and Echo). Where Trigger used to start of sending signals, and Echo will receive signals. Step2: Specify start and end time as the following: StartTime=Current time StopTime=Current time Step3: While (Echo==0)

StartTime=Current time

While (Echo==1)

StopTime=Current time

Step4: Find difference between two times use the following equation:

$$Times = StopTime - StartTime$$

Step5: Multiply with sonic speed (34300 cm/s) and divided by 2 because there send and back:

$$Distance = (Times * 34300)/2$$

End

The surveillance camera is connected to the main controller directly via the camera slot located in Raspberry Pi4. Figure (3.4) illustrate camera connection. Algorithm (3.2) camera work.

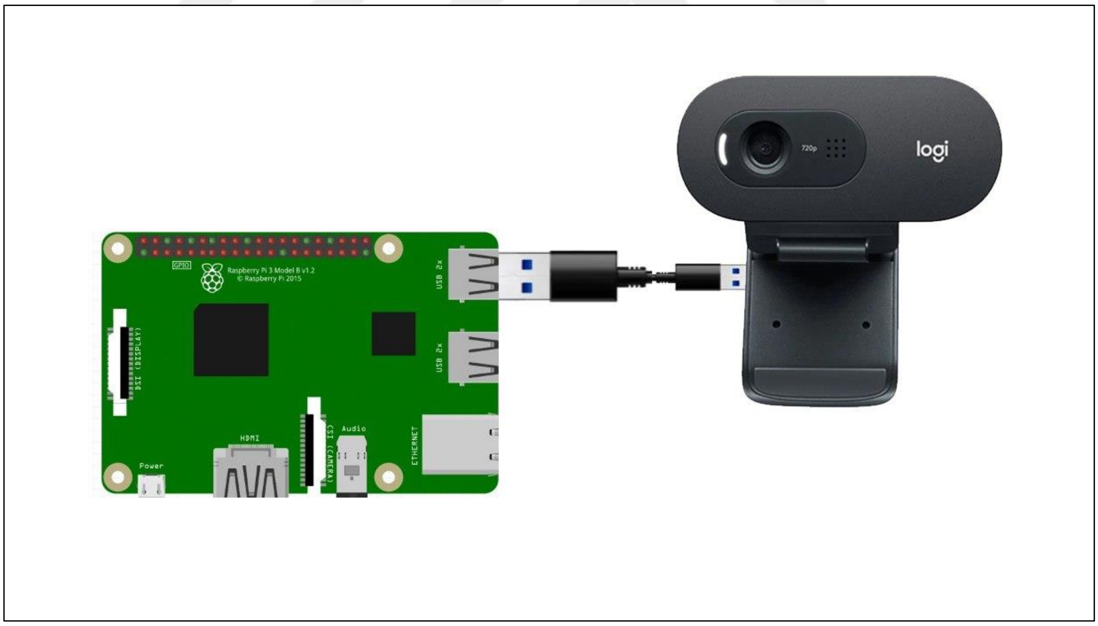


Figure 3.4 Camera connection

Algorithm (3.2): Camera Work
Input: Distance
Output: Capturing
Begin: Step1: Read distance from ultrasonic sensor Step2: If the distance is less than 100 cm, the camera will open and capturing video 5 second only then close. Step3: If the distance is more than 100 cm the camera still close. End

The complete electronic circuit can be illustrated by the Figure (3.5).

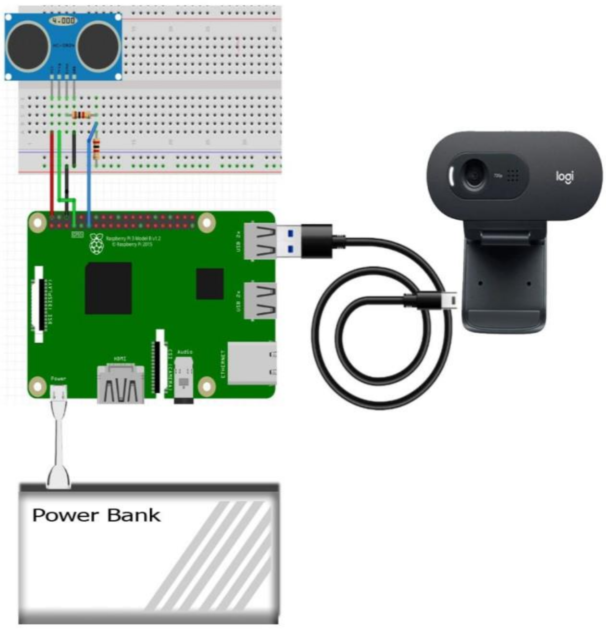


Figure 3.5 Complete electronic circuit

3.5 Detection Stage

Face detection is a very important process. In this section of the proposed system the HAAR cascading algorithm developed for the purpose of detecting faces from the image will be explained Figure (3.6).

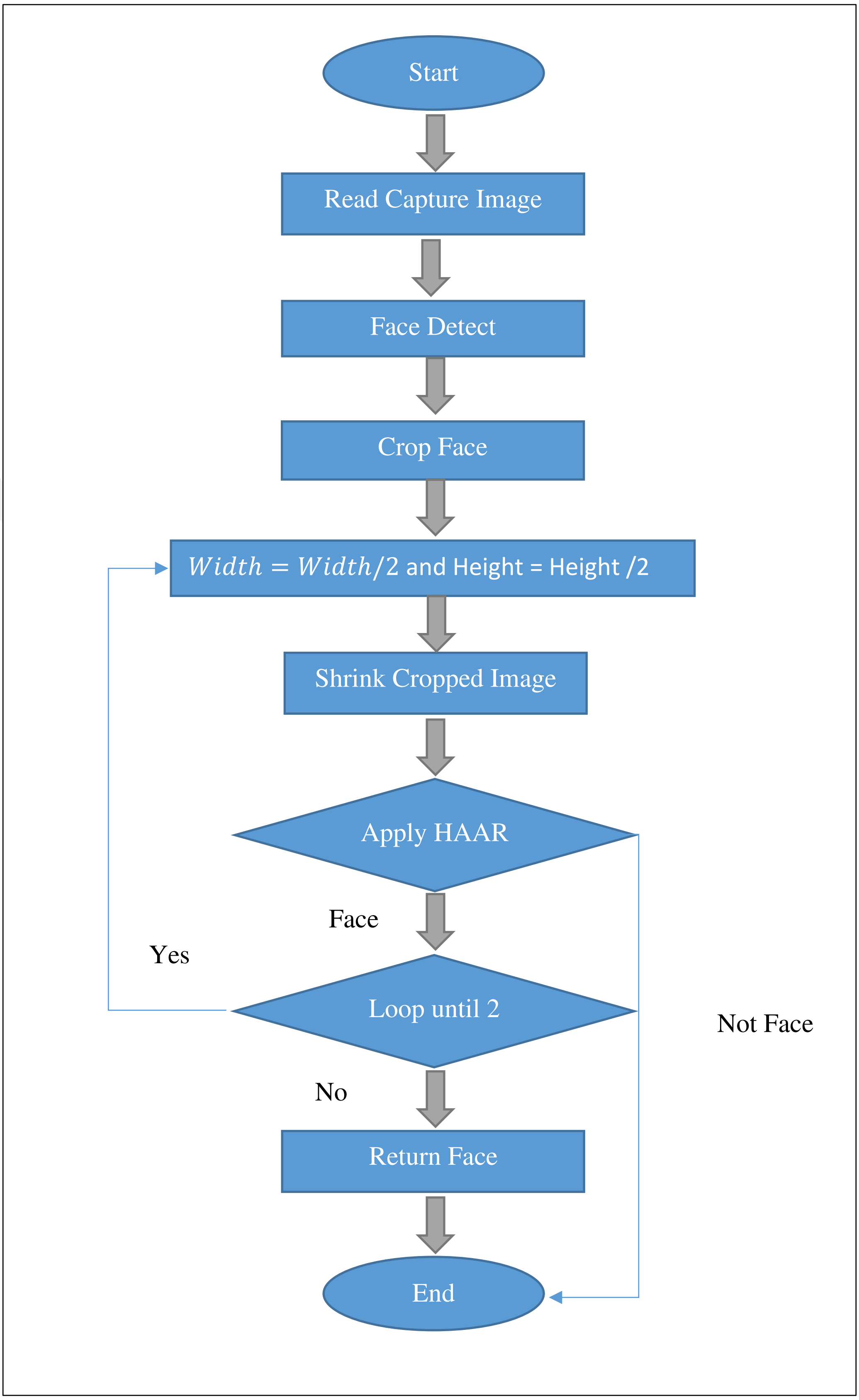


Figure 3.6 Modified HAAR Cascading Algorithm

The HAAR cascading algorithm is one of the common algorithms in detecting faces because of the speed of its performance, but it is not very accurate in detecting faces. The algorithm was developed to be used within Raspberry Pi for the purpose of detecting people's faces within the scope of surveillance. Algorithm (3.3) illustrate the modification of HAAR cascading algorithm.

Algorithm (3.3): Modification HAAR Cascading
Input: Captures Image
Output: Face Detection
Begin: Step1: Read capture image Step2: Apply HAAR cascading for detecting faces. Step3: Cropping face from image Step4: For i from 0 to 2 A. Divided the width and height of cropping face by 2. $$\boldsymbol{Width_{new}} = \frac{\boldsymbol{Width_{old}}}{\mathbf{2}} \quad (\mathbf{3.1})$$ $$\boldsymbol{Height_{new}} = \frac{\boldsymbol{Height_{old}}}{\mathbf{2}} \quad (\mathbf{3.2})$$ B. Shrink the cropping face with new dimensions $(Width_{new}, Height_{new})$. C. Apply HAAR cascading. Step5: Check if the result of HAAR is face in three times then return face otherwise return not face. End

Where the modified HAAR cascading algorithm checks the face three times, if the result is a face in the three times, it means that it is a face, otherwise it is not a face. Figure (3.6) illustrate the modification HAAR cascading algorithm works.

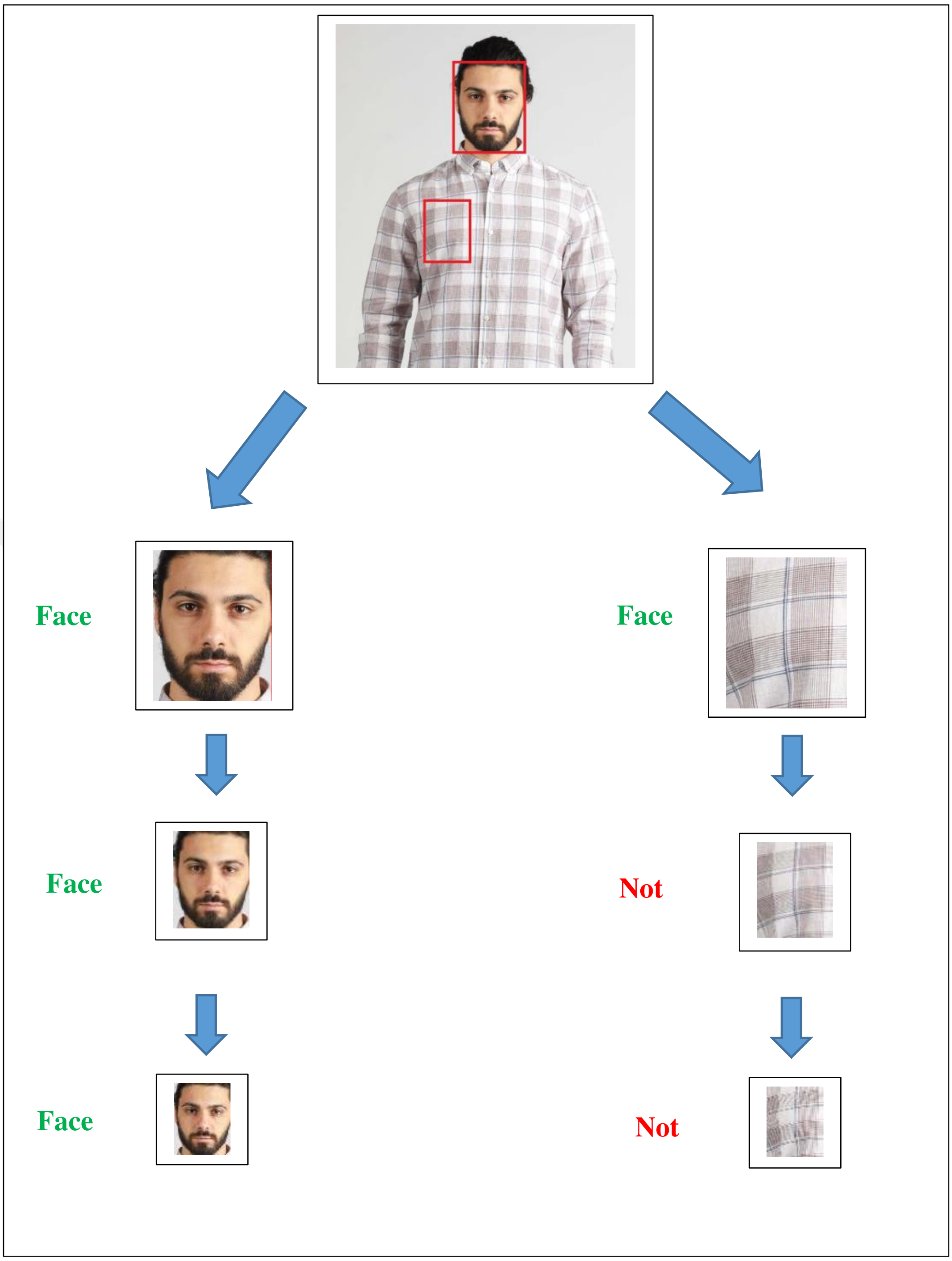


Figure 3.6 Modification HAAR work

3.6 Recognition Stage

The proposed device takes pictures, detects the face, and then sends the crop image to the server, which is the last stage of the proposed system, which in turn identifies the person. Figure (3.7) illustrate flowchart of recognition stage.

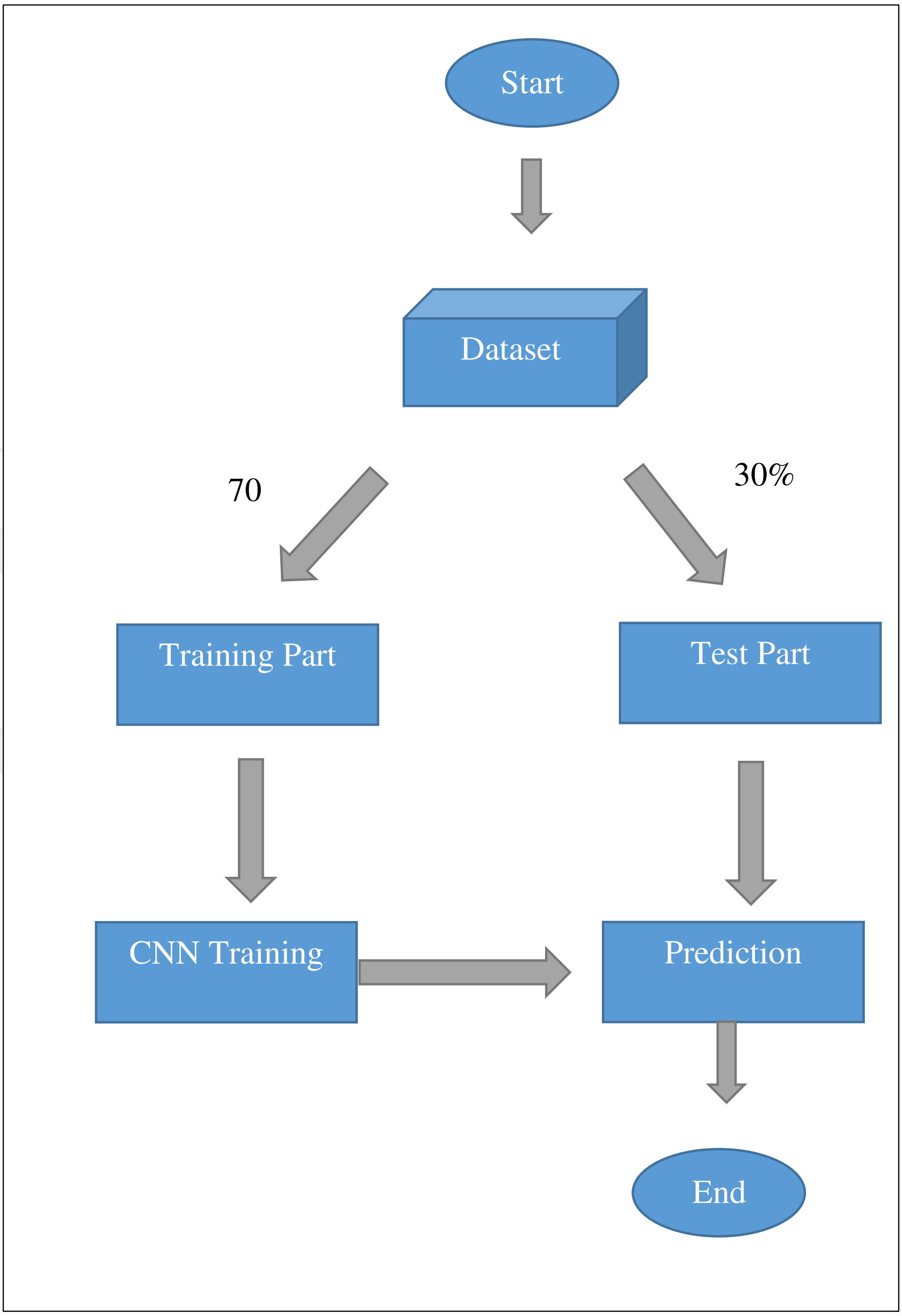


Figure 3.7 Recognition stage flowchart

After the faces are cut out of all the data images, they are divided into two parts, 70 percent of which are used to train the proposed model of the CNN algorithm and 30 percent are used for testing. Figure (3.8) illustrate steps of CNN training.

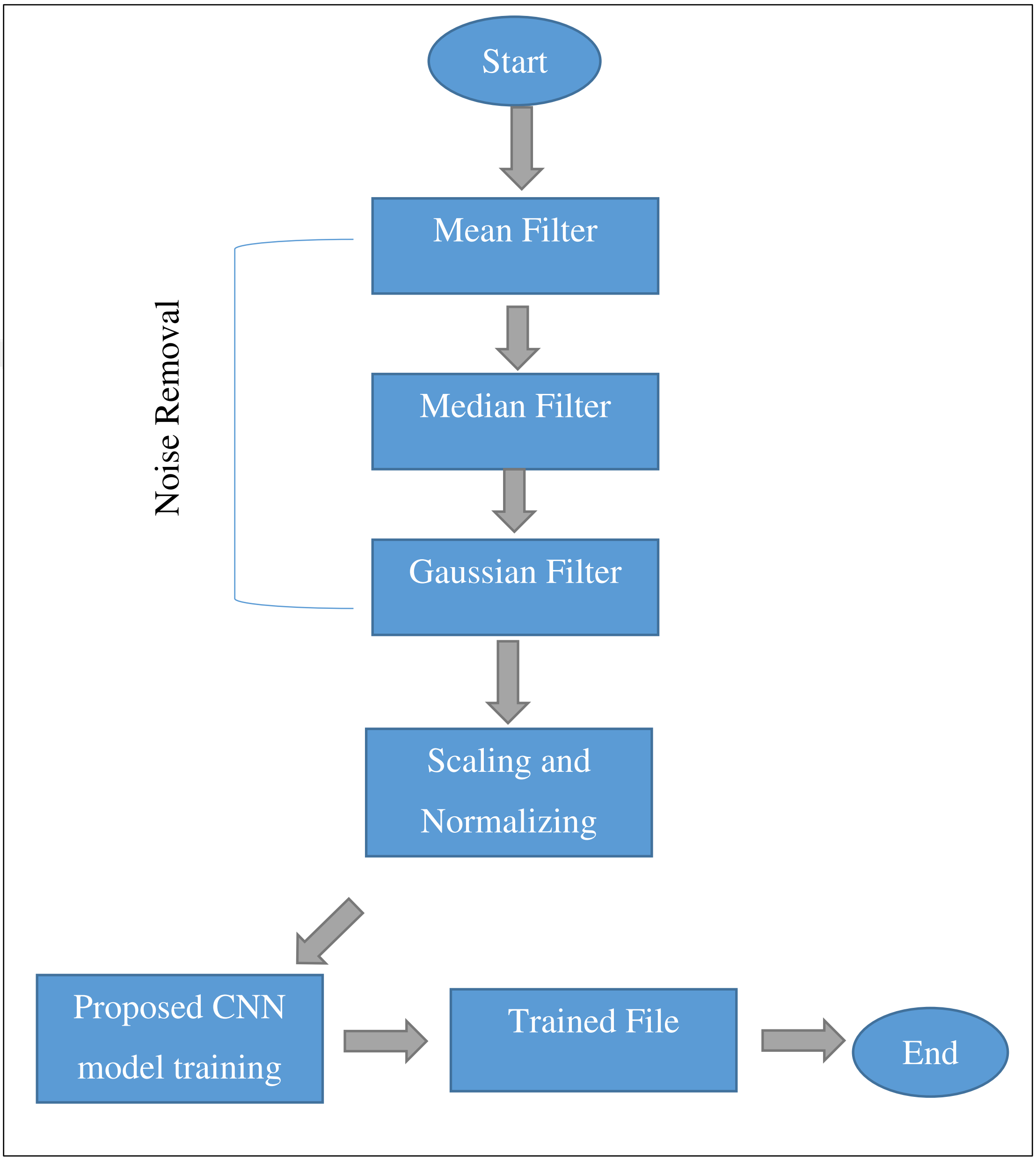


Figure 3.8 CNN training steps

First, noise is removed from images using three types of filters, namely (mean, median, and Gaussian) where each type of these filters removes a specific type of noise in digital images. Algorithm (3.4) illustrate noise removal steps.

Algorithm (3.4): Noises Removal
Input: Cropping Face
Output: Image Enhancement
Begin: Step1: Read cropping face image Step2: Apply mean filter to remove intensity variation between one pixel and the next by find the average between surrounding pixels. Step3: Apply median filter to remove salt and pepper noise by finding the mid value from surrounding pixels. Step4: Apply Gaussian filter for non-uniform noise reduction. Step5: Return enhancement image. End

After the noise is removed from the image, the process of scaling and normalizing the image begins. The goal of these steps is to reduce the size of the data for the purpose of increasing the speed of system performance. Algorithm (3.5) illustrate scaling and normalizing steps.

Algorithm (3.5): Scaling and Normalizing
Input: Enhanced images
Output: Normalized images
Begin: Step1: Read dataset images Step2: Shrink the images with size (64×64) experimentally Step3: Convert all pixel value to floats. Step4: Apply normalizing equation:

$$\boldsymbol{Pixel_{i,j}} = \frac{\boldsymbol{Pixel_{i,j}}}{\boldsymbol{255}} \qquad (\boldsymbol{3.3})$$

Step5: Return normalizing image

End

After the data images are prepared, they are trained using the proposed model of the CNN algorithm. Where a model consisting of 11 layers of the CNN algorithm was designed, which is as shown in the Figure (3.9). where the proposed model consists of three convolution layers, three pooling layers, two dropout layers, two fully connection layers, and one SoftMax layer. In the proposed model, a number of convolution layers are used, because the features of the faces may be close and therefore more convolution layers must be used in extracting features to make the proposed system capable to identify people. And when the number of people increases significantly, the number of layers must also be increased. After each convolution layer, a pool layer must be placed, because the features that extracted by convolution layer are random, so it must be filtered and the best features between them must be chosen, and here comes the role of the pool layer in choosing the best features. With regard to dropout layers, there are some features that are extracted that may be strong and distinctive, but they may negatively affect the facial recognition algorithm during training, and therefore they must be delete. The number of dropout layers is related to the number of fully connection layers. The SoftMax classifier is considered one of the best works that have been used with the CNN algorithm, because its value ranges between negative 1 and 1, and this gives it greater freedom in classifying classes.

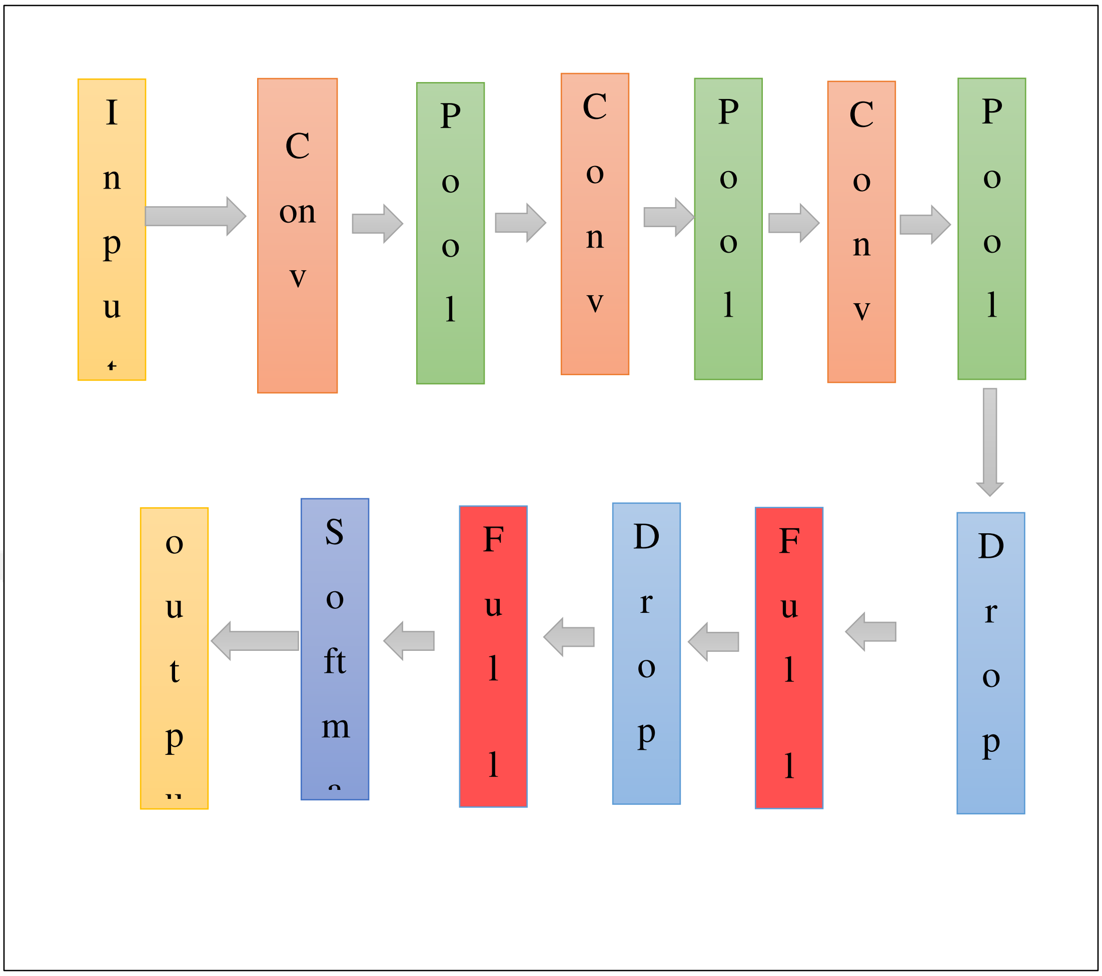


Figure 3.9 Proposed CNN model

The proposed model contains three convolution layers for extract big numbers of features from image in additional to used RelU as activation function to prevent extracted features to be neared from zero. Table (3.1) illustrate specification of used convolution layers.

Table 3.1 Convolution layer specification

Layer	Kernel Size	Activated Function
Conv.1	64	RelU
Conv.2	32	RelU
Conv.3	32	RelU

After extract features from image pooling layer will select the best features from features map, Table (3.2) illustrate pooling layers specification.

Table 3.2 Pooling layer specification

Layer	Kernel Size
Pool.1	7
Pool.2	5
Pool.3	3

The dropout layer used to remove nodes from fully connection layer that not effect on the prediction. Table (3.3) illustrate dropout layer specification.

Table 3.3 Dropout layer specification

Layer	Drop Size
Drop.1	0.45
Drop.2	0.25

Fully connection layers are the last layers in proposed model, these layers are used with SoftMax to face recognition. Table (3.4) illustrate fully connection layer specification.

Table 3.4 Full connection layer specification

Layer	Activated Function
Full.1	RelU
Full.2	SoftMax

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

Face detection and recognition are two essential tasks in computer vision that involve detecting human faces in images or videos and identifying the individuals present in those faces. These tasks have numerous applications, including facial authentication, surveillance. It's important to note that while face detection and face recognition are related tasks, they serve different purposes. Face detection is the initial step of locating faces in an image, and face recognition follows to determine the identity of the detected faces. In this chapter, the most important results will be clarified by proposing a system for monitoring, facial recognition and identification of people.

4.2 Datasets Explanation

In the proposed system, two datasets were relied upon, one (Face Recognition Dataset) and the other built in a real way. The purpose of using more than one dataset containing faces was for the purpose of testing the accuracy of the proposed system in detecting faces and recognizing people through the proposed and developed techniques. Figure (4.1) illustrate samples of (Face Recognition Dataset), Figure (4.2) illustrate samples of real dataset.

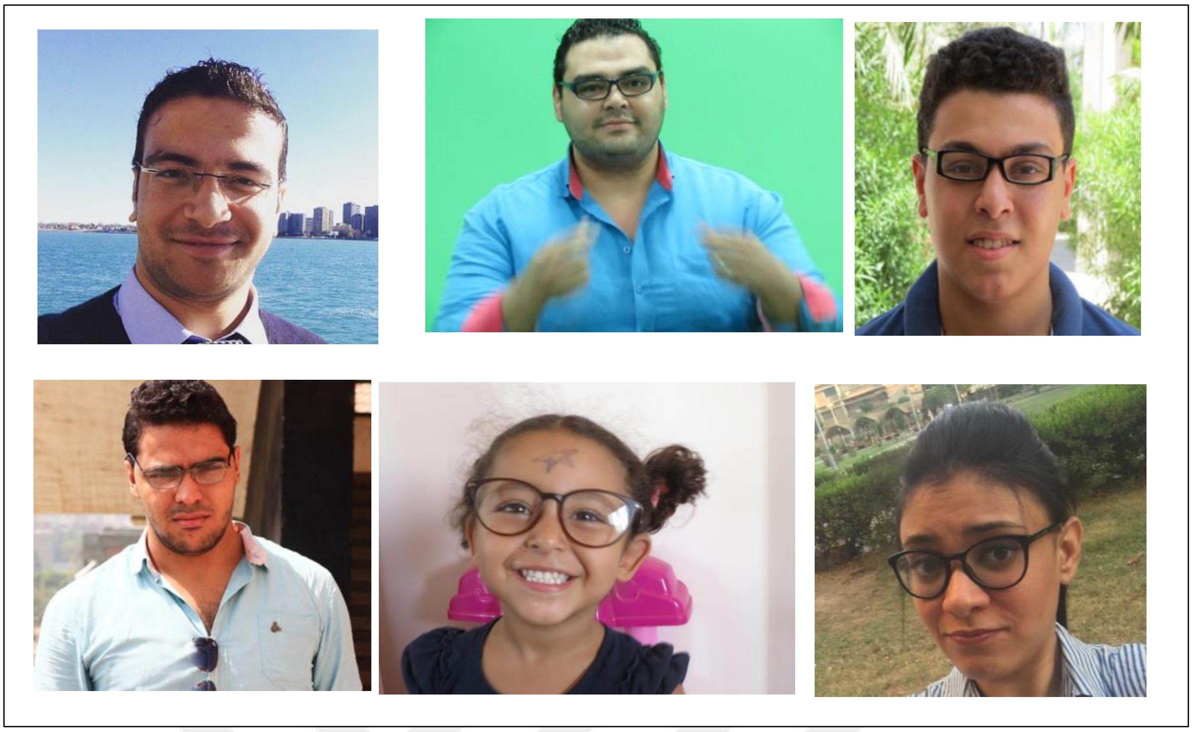


Figure 4.1 Samples of (Face Recognition Dataset)

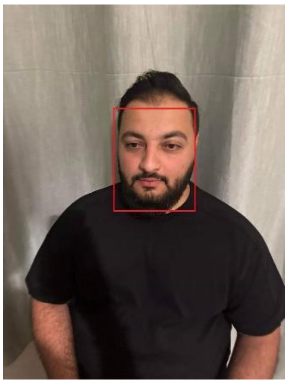
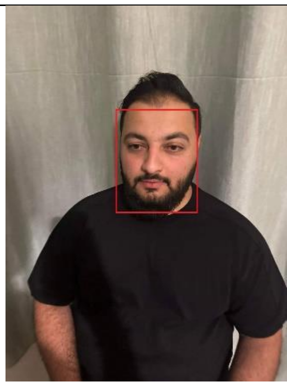
Figure 4.2 Samples of real dataset

4.3 Face Detection Results

Face detection is the process of locating human faces in images or video frames. It involves identifying the coordinates (bounding box) of the regions that contain faces. Where the HAAR cascading algorithm was developed for the purpose of detecting faces more accurately. Table (4.1) illustrate results of face detection using modified HAAR cascading algorithm.

Table 4.1 Comparison between Classical and modified HAAR cascading results for face detection

Sample No.	Classical HAAR Cascading Face Detection	Modified HAAR Cascading Face Detection
1		
2		

3		
4		
5		

It can be seen from the previous table that some images when using the classical HAAR cascading algorithm have identified two faces in the image despite the presence of one person in the image as in the second and fifth samples. When the modified HAAR cascading algorithm was used, the parts that did not represent a face were removed and the accuracy of the face was better.

On two databases, a modification was made to HAAR's algorithm, which proved to increase the accuracy of face identification in images with repeated faces and to speed the processing time of images. Tables (4.2) and (4.3) show the results of the testing on the modified HAAR algorithm.

Table 4.2 HAAR cascading face detection results using real dataset

Technique	Classical HAAR	Modified HAAR
Number of Image (Real Dataset)	150	150
Success detection	131	148
Failed detection	19	2
Accuracy	87.333	98.667

Table 4.3 HAAR cascading face detection results using (Face Recognition Dataset)

Technique	Classical HAAR	Modified HAAR
Number of Image (downloading Dataset)	320	320
Success detection	276	309
Failed detection	44	11
Accuracy	86.250	96.563

4.4 Cropping Results

The process of cropping the face from the image is one of the important processes in the task of identifying people, where the important part of the image is cropped and the rest of the parts of the image are deleted to be sent to the server so that it can identify the person. In addition, the process of cropping part of the image, so the size of the image is less than the original image, and therefore the process of sending it over the network is faster. Figure (4.3) illustrate results of cropping process.

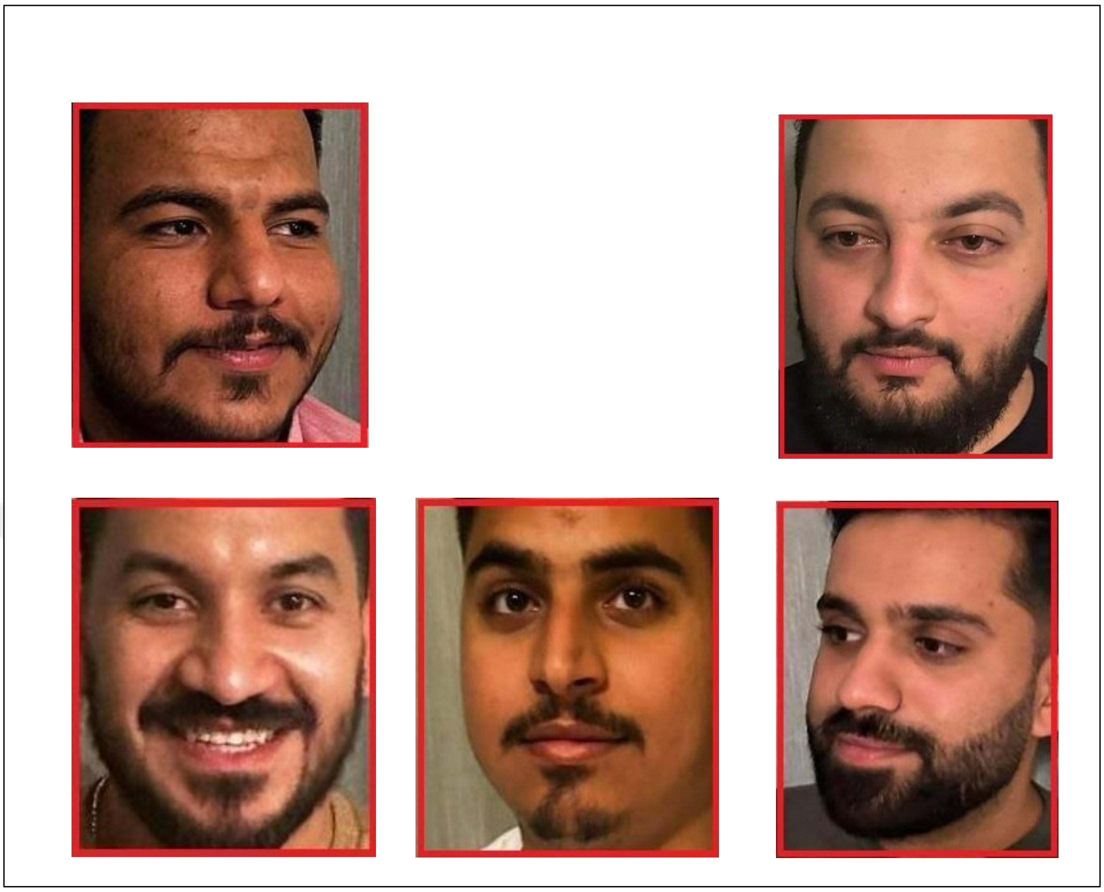


Figure 4.3 Cropping results

4.5 Face Recognition Results

The process of recognizing faces is important in the system of surveillance, as during this section many practical experiments were conducted for the purpose of training the proposed model of the CNN algorithm, and many important and high results were reached in prediction by changing the number of training epochs and the percentage of dataset splitting. There are three cases that have been tested and they are as follows:

Case1: in first case split dataset into 90% of images for training and 10% of images for testing for two datasets (Face Recognition Dataset and real dataset). Table (4.4) illustrate results of train-test of real dataset, and Table (4.5) illustrate results of train-test of Face Recognition Dataset.

Table 4.4 Train-test of real dataset of case1 for face recognition

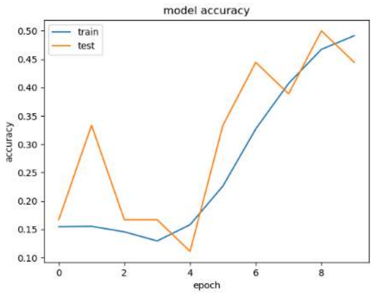
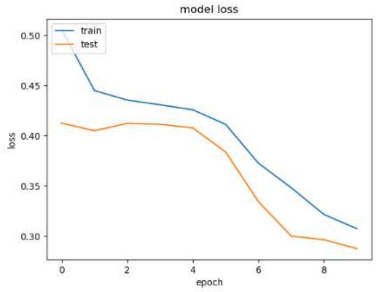
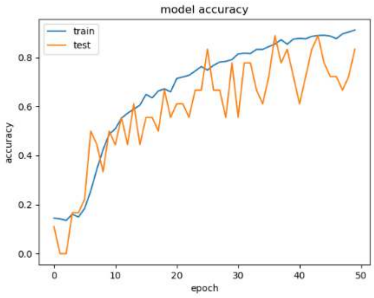
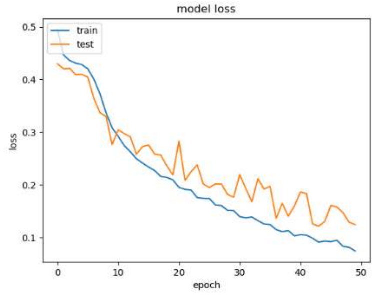
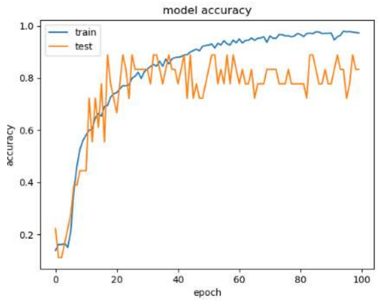
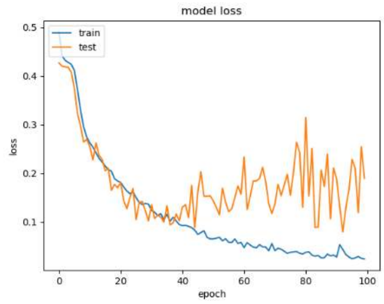
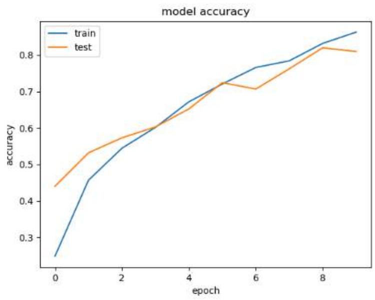
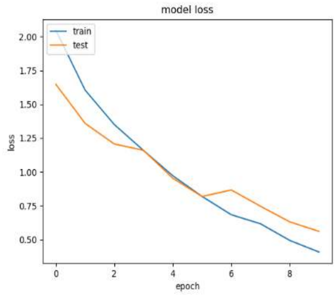
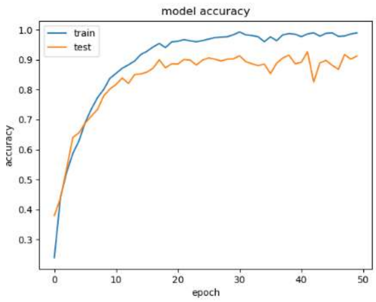
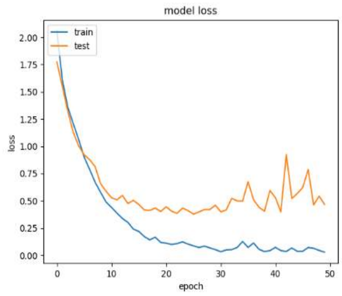
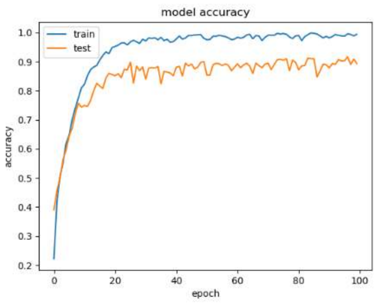
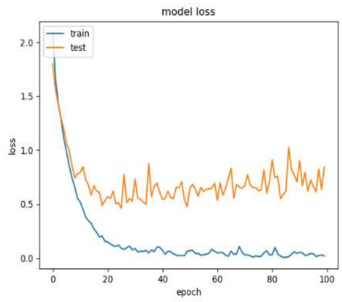
Epochs	Train-test accuracy	Train-test loss
10		
50		
100		

Table 4.5 Train-test of Face Recognition Dataset of case1 for face recognition

Epochs	Train-test accuracy	Train-test loss
10		
50		
100		

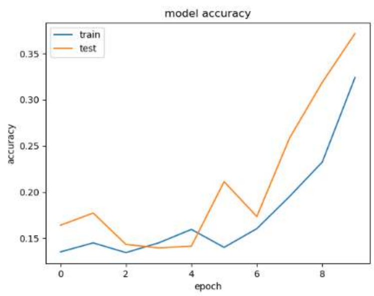
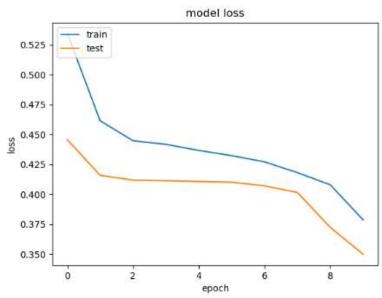
The previous results can be summarized in the Table (4.6) where the highest accuracy in the forecast is 0.8333 for real dataset and 0.8976 for Face Recognition Dataset.

Table 4.6 Summarized of case1 results for face recognition

Epoch	Dataset	Train-ACC	Train-Loss	Test-ACC	Test-Loss
10	Real	0.4914	0.3074	0.4444	0.2876
	Face Recognition Dataset	0.8464	0.4472	0.7901	0.6868
50	Real	0.9118	0.0745	0.8333	0.1244
	Face Recognition Dataset	0.9868	0.0474	0.9177	0.3826
100	Real	0.9731	0.0244	0.8333	0.1900
	Face Recognition Dataset	0.9917	0.0198	0.8976	0.8731

Case2: in second case split dataset into 80% of images for training and 20% of images for testing for two datasets (Face Recognition Dataset and real dataset). Table (4.7) illustrate results of train-test of real dataset, and Table (4.8) illustrate results of train-test of Face Recognition Dataset.

Table 4.7 Train-test of real dataset of case2 for face recognition

Epochs	Train-test accuracy	Train-test loss
10		

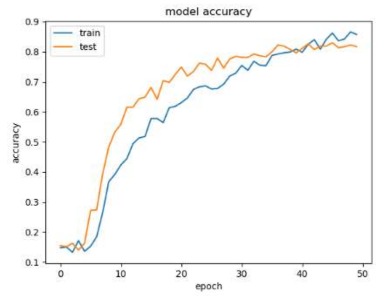
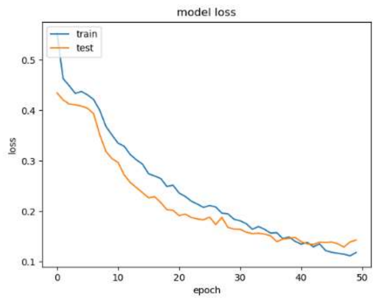
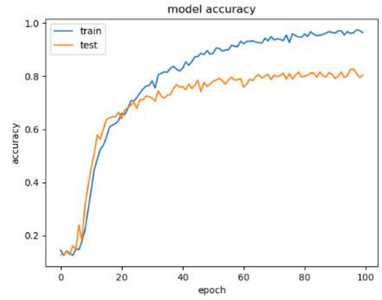
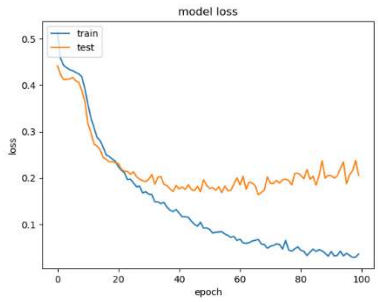
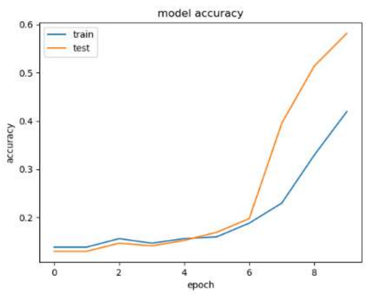
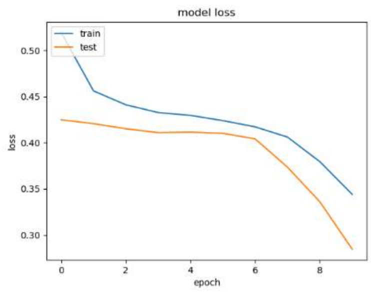
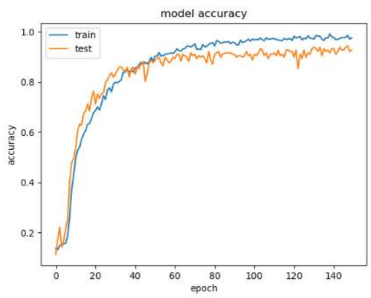
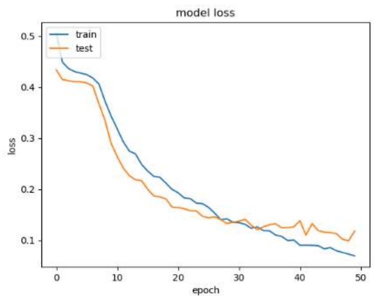
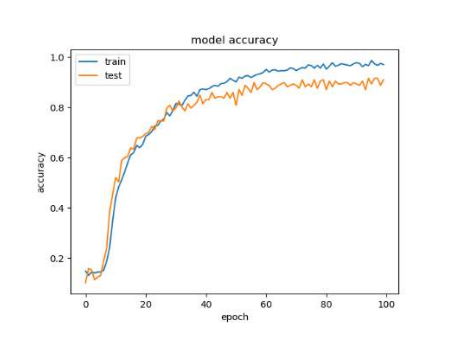
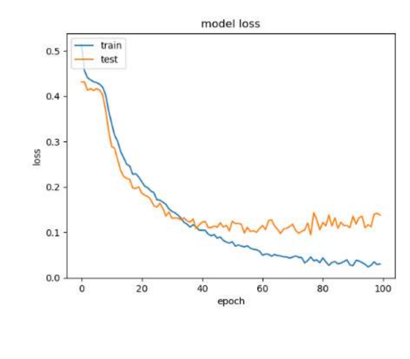
50		
100		

Table 4.8 Train-test of Face Recognition Dataset of case2 for face recognition

Epochs	Train-test accuracy	Train-test loss
10		
50		

100		

The previous results can be summarized in the Table (4.9) where the highest accuracy in the forecast is 0.8438 for real dataset and 0.9096 for Face Recognition Dataset.

Table 4.9 Summarized of case2 results for face recognition

Epoch	Dataset	Train-ACC	Train-Loss	Test-ACC	Test-Loss
10	Real	0.3241	0.3787	0.3717	0.3500
	Face Recognition Dataset	0.4197	0.3442	0.5819	0.2848
50	Real	0.8647	0.1447	0.8158	0.1874
	Face Recognition Dataset	0.9168	0.0695	0.8757	0.1182
100	Real	0.9643	0.0363	0.8438	0.2059
	Face Recognition Dataset	0.9698	0.0305	0.9096	0.1378

Case3: in third case split dataset into 70% of images for training and 30% of images for testing for two datasets (Face Recognition Dataset and real dataset). Table (4.10) illustrate results of train-test of real dataset, and Table (4.11) illustrate results of train-test of Face Recognition Dataset.

Table 4.10 Train-test of real dataset of case3 for face recognition

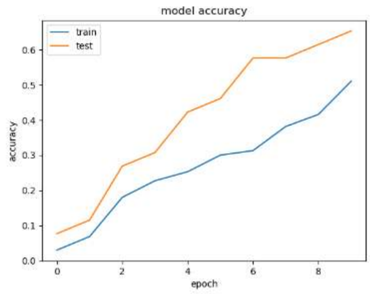
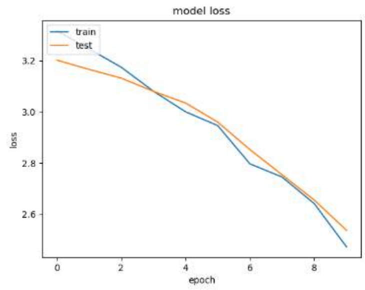
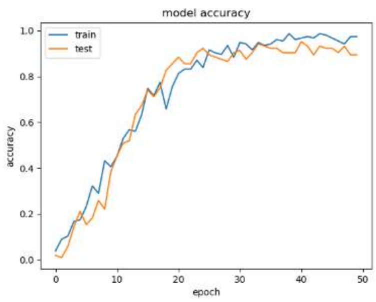
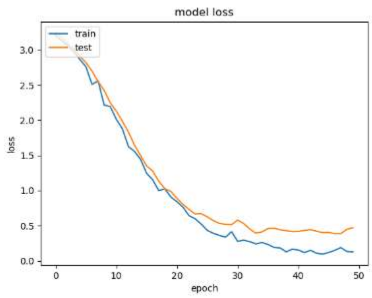
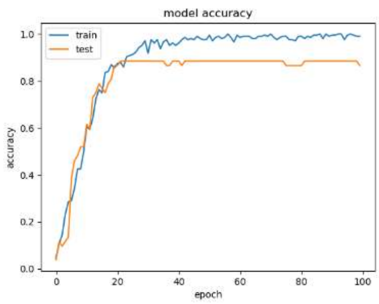
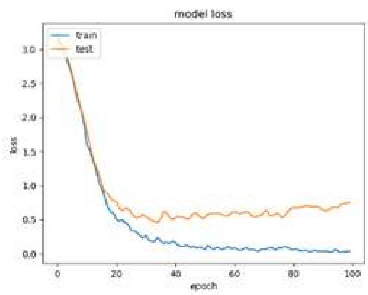
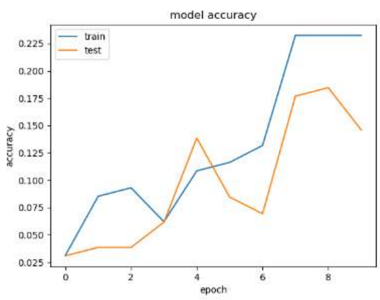
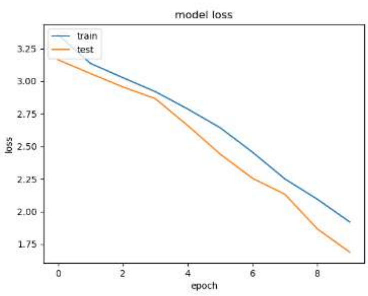
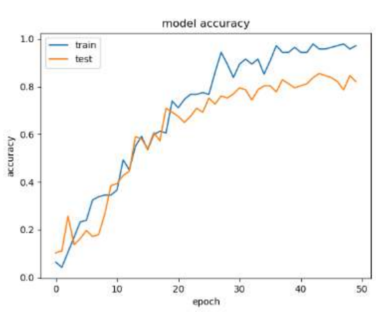
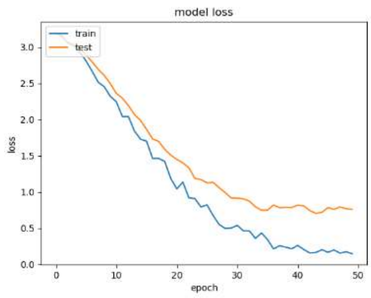
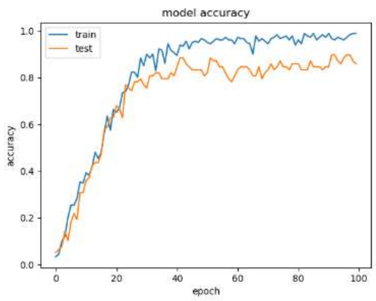
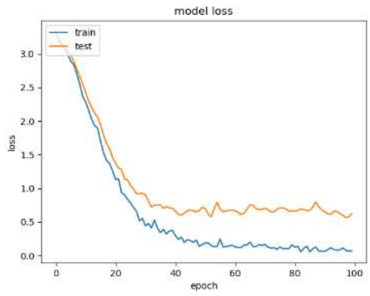
Epochs	Train-test accuracy	Train-test loss
10		
50		
100		

Table 4.11 Train-test of Face Recognition Dataset of case3 for face recognition

Epochs	Train-test accuracy	Train-test loss
10		

50		
100		

The previous results can be summarized in the Table (4.12) where the highest accuracy in the forecast is 0.8742 for real dataset and 0.9447 for Face Recognition Dataset.

Table 4.12 Summarized of case3 results for face recognition

Epoch	Dataset	Train-ACC	Train-Loss	Test-ACC	Test-Loss
10	Real	0.3568	0.2241	0.4012	0.2547
	Face Recognition Dataset	0.1510	0.1984	0.2254	0.1755
50	Real	0.9564	0.1145	0.8795	0.4995
	Face Recognition Dataset	0.9643	0.1475	0.8214	0.5247
100	Real	0.9742	0.1252	0.8742	0.6705

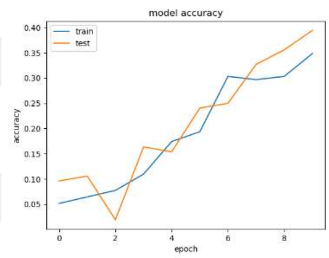
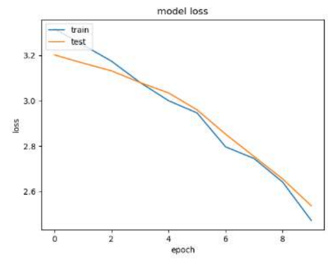
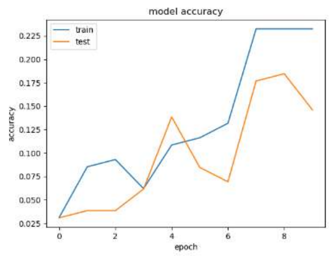
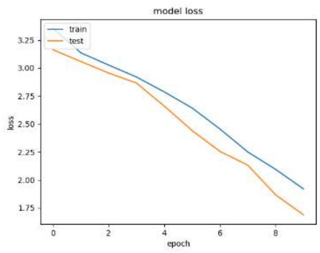
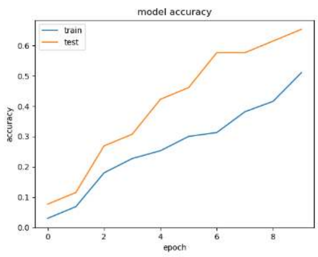
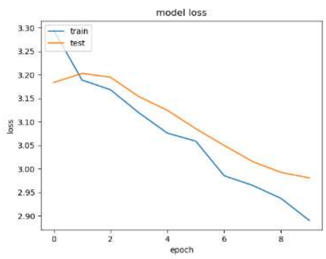
	Face Recognition Dataset	0.9818	0.1463	0.9447	0.1784

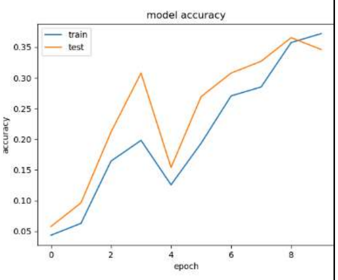
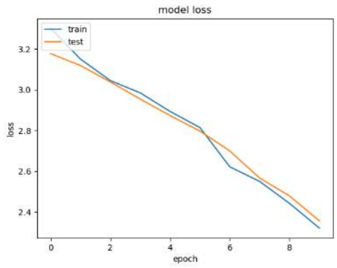
4.6 Comparison CNN Results

In this section, the results of the CNN algorithm will be discussed before and after using the modified HAAR algorithm. Three cases have been taken for comparison and they are as follows:

Case I: When using 10 iterations, Table (4.13) illustrate results of training and testing.

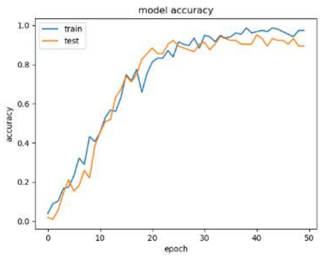
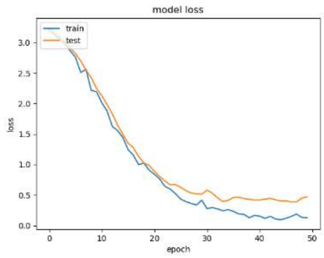
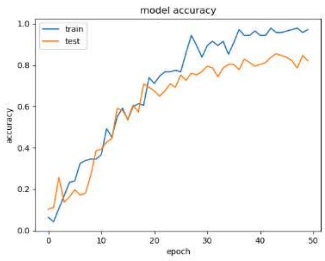
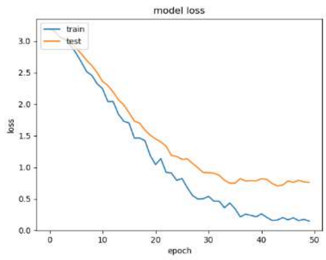
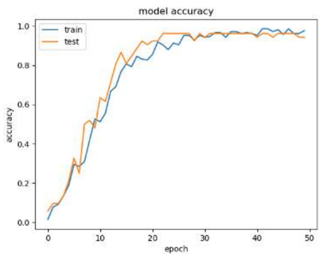
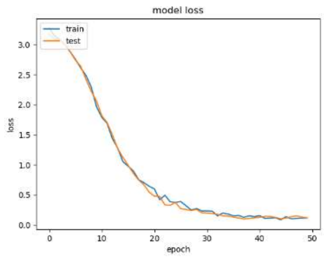
Table 4.13 Results of training and testing by 10 iterations for face recognition

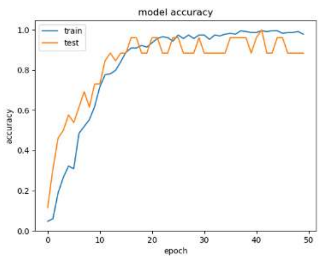
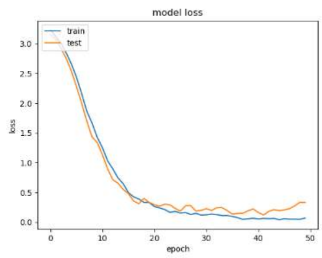
Algorithm	Dataset	Accuracy	Loss
Classical Haar Classifier +CNN	Real Dataset		
	Face Recognition Dataset		
Modified Haar Classifier +CNN	Real Dataset		

	Face Recognition Dataset		

Case II: When using 50 iterations for training, Table (4.14) illustrate results of 50 iterations.

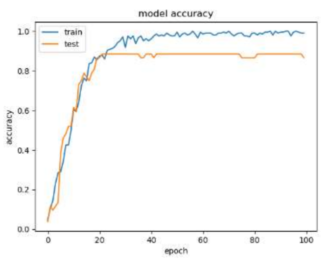
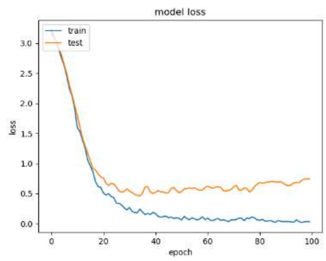
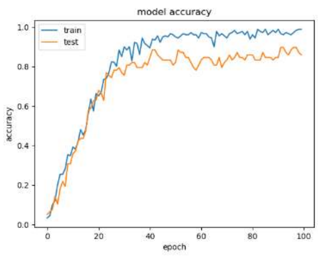
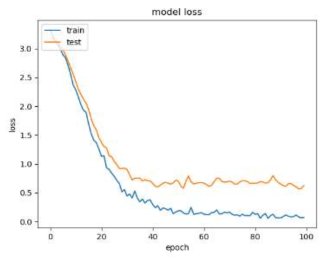
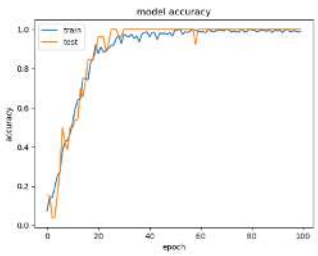
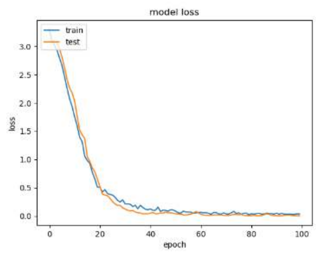
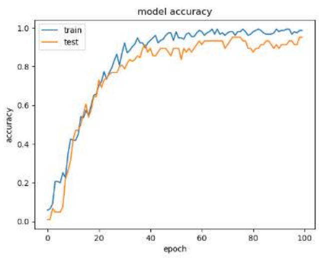
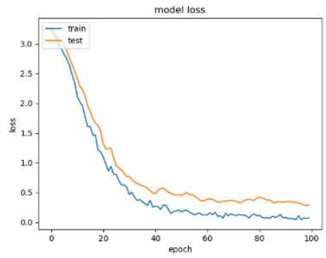
Table 4.14 Results of training and testing by 50 iterations for face recognition

Algorithm	Dataset	Accuracy	Loss
Classical Haar Classifier +CNN	Real Dataset		
	Face Recognition Dataset		
Modified Haar Classifier +CNN	Real Dataset		

	Face Recognition Dataset		

Case III: When using 100 iterations for training, Table (4.15) illustrate results of 100 iterations.

Table 4.15 Results of training and testing by 100 iterations for face recognition

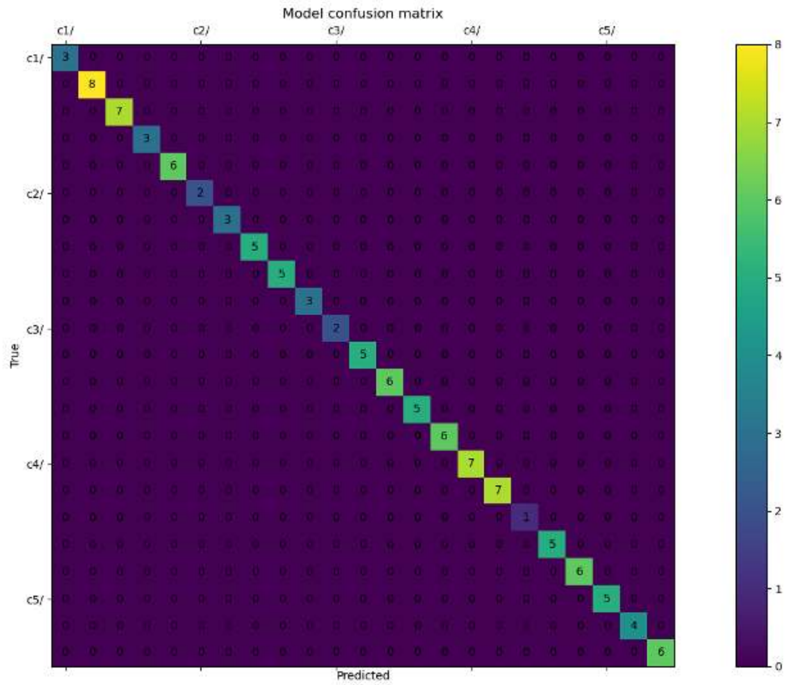
Algorithm	Dataset	Accuracy	Loss
Classical Haar Classifier +CNN	Real Dataset		
	Face Recognition Dataset		
Modified Haar Classifier +CNN	Real Dataset		
	Face Recognition Dataset		

Through many experiments notice that the accuracy has reached the best possible in the hybrid system in iteration 100 and the results can be summarized in the Table (4.16).

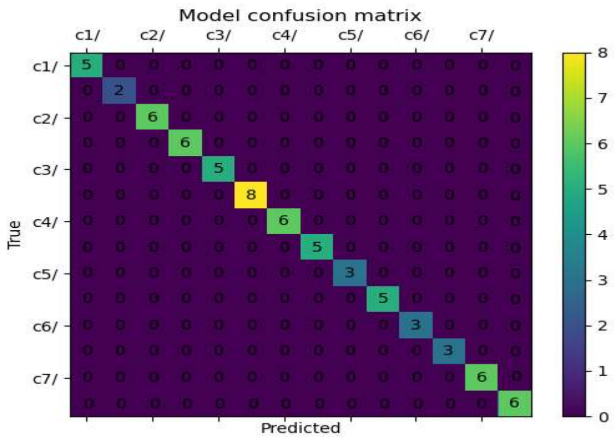
Table 4.16 Results summarized for face recognition

Iteration	Dataset	Algorithm	Train Accuracy	Train Loss	Test Accuracy	Test Loss
10	Real	Classical Haar Classifier + CNN	0.3568	0.2241	0.4012	0.2547
	Face Recognition Dataset		0.1510	0.1984	0.2254	0.1755
	Real	Modified Haar Classifier + CNN	0.4982	0.2936	0.6963	03122
	Face Recognition Dataset		0.3641	0.2357	0.3417	0.2411
50	Real	Classical Haar Classifier + CNN	0.9564	0.1145	0.8795	0.4995
	Face Recognition Dataset		0.9643	0.1475	0.8214	0.5247
	Real	Modified Haar Classifier +CNN	0.9874	0.1865	0.9677	0.1995
	Face Recognition Dataset		0.9748	0.1479	0.8641	0.1983
100	Real	Classical Haar Classifier + CNN	0.9742	0.1252	0.8742	0.6705
	Face Recognition Dataset		0.9818	0.1463	0.9447	0.1784
	Real	Modified Haar Classifier +CNN	0.9871	0.0346	**0.9968**	0.0020
	Face Recognition Dataset		0.9858	0.1177	**0.9623**	0.2190

The prediction accuracy reached the maximum degree in the number of 100 iterations, which is 99.68 for the real database and 96.23 for Face Recognition Dataset. Figure (4.4) illustrate confusion matrix, Table (4.17) illustrate system performance.



(A)



(B)

Figure 4.4 Confusion matrix (A) Face recognition dataset (B) Real Dataset

Table (4.17) System performance

Dataset	Class	Precision	Recall	F1-Score
Face Recognition Dataset	Class 1	1.00	1.00	1.00
	Class 2	1.00	1.00	1.00
	Class 3	1.00	0.97	0.98
	Class 4	0.94	0.94	0.94
	Class 5	1.00	1.00	1.00
	….	….	….	….
	Average	0.9623	0.9694	0.9658
Real Dataset	Class 1	0.96	1.00	0.98
	Class 2	1.00	1.00	1.00
	Class 3	1.00	1.00	1.00
	Class 4	1.00	0.94	0.97
	Class 5	0.98	0.98	0.98
	….	….	….	….
	Average	0.9968	0.9961	0.9973

4.7 Interface and Implementation

Several tested were applied on the proposed system, this section will display user interface of proposed system. At first, Figure (4.5) illustrates the main interface of the proposed system.

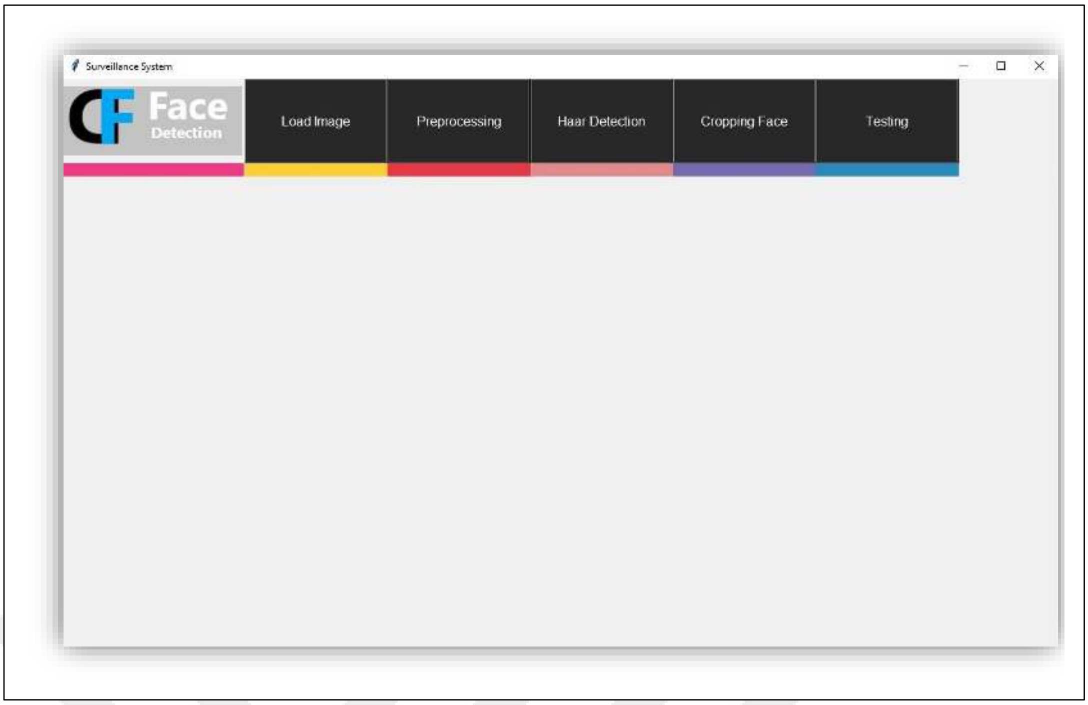


Figure 4.5 The main interface

The modify HAAR cascading algorithm is used for face detection. Figure (4.6) illustrates face detection.

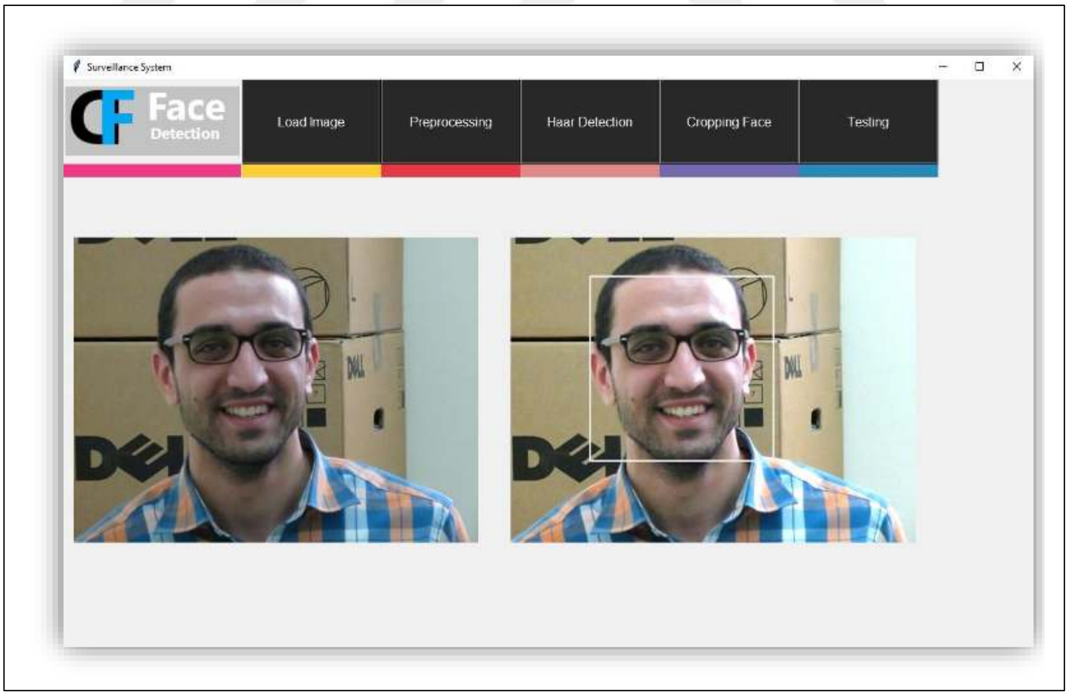


Figure 4.6 face detection using the modify HAAR cascading

The image is cropped after detection as n Figure (4.7).

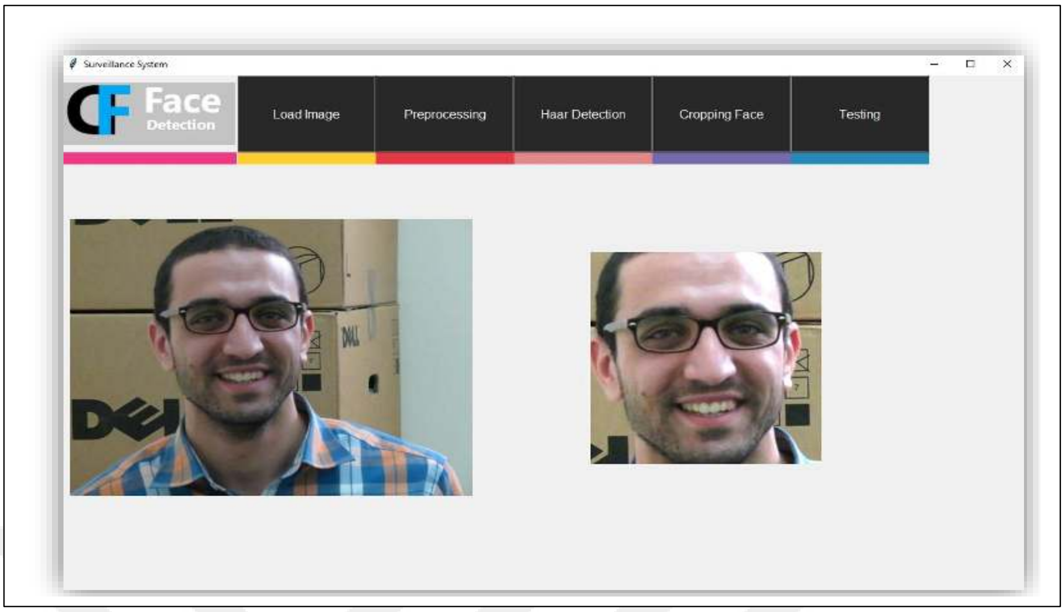


Figure 4.7 image after cropping faces

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In this thesis, a system was proposed to recognize faces in addition to distinguishing between them, and through experiments conclude the following:

1. Concluding that shrinking the image several times and testing it indicates that the HAAR cascading algorithm has decreased the error rate. As shown in Table (4.1).
2. For real dataset the best accuracy of face detection is 87.333 for classical HAAR and 98.667 for modified HAAR as in Table (4.2).
3. For downloaded dataset the best accuracy of face detection is 86.250 for classical HAAR and 96.563 for modified HAAR as in Table (4.3).
4. Another conclusion after proposing a CNN model is that the highest accuracy of its interface to classify faces is at 100 iterations.
5. From Table (4.12) we can conclude that the best accuracy is 0.8742 for the real dataset and 0.9447 for the downloaded dataset.
6. Additionally, it can be concluded that using modified HAAR with CNN achieves high accuracy. The results show 99.68% accuracy for the real database and 96.23% for the face recognition dataset, as indicated in Table (4.16).

5.2 Recommendations

The monitoring system can be developed using rapid implementation techniques, and there are many recommendations, which are as follows:

1. The face detection algorithm can be developed using the YOLO algorithm being faster in identifying the face.
2. The system can be developed using a network of cameras instead of a single camera.

REFERENCES

[1] Wilmer, Jeremy B., et al. "Human face recognition ability is specific and highly heritable." *Proceedings of the National Academy of sciences* 107.11 (2010): 5238-5241.

[2] Zhao, Zhong-Qiu, De-Shuang Huang, and Bing-Yu Sun. "Human face recognition based on multi-features using neural networks committee." *Pattern recognition letters* 25.12 (2004): 1351-1358.

[3] Barnouti, Nawaf Hazim, Sinan Sameer Mahmood Al-Dabbagh, and Wael Esam Matti. "Face recognition: A literature review." *International Journal of Applied Information Systems* 11.4 (2016): 21-31.

[4] Mohammed, Abdul Adeel, et al. "Human face recognition based on multidimensional PCA and extreme learning machine." *Pattern recognition* 44.10-11 (2011): 2588-2597.

[5] Haddadnia, Javad, and Majid Ahmadi. "N-feature neural network human face recognition." *Image and Vision Computing* 22.12 (2004): 1071-1082.

[6] Li, Shutao, et al. "Deep learning for hyperspectral image classification: An overview." *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019): 6690-6709.

[7] Mikołajczyk, Agnieszka, and Michał Grochowski. "Data augmentation for improving deep learning in image classification problem." *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, 2018.sa

[8] Pak, Myeongsuk, and Sanghoon Kim. "A review of deep learning in image recognition." *2017 4th international conference on computer applications and information processing technology (CAIPT)*. IEEE, 2017.

[9] Sangaiah, Kumar, Mohamed Elhoseny, and Sung Wook Baik. "Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities." (2017).

[10] Khan, Muhammad Zeeshan, et al. "Deep unified model for face recognition based on convolution neural network and edge computing." IEEE Access 7 (2019): 72622-72633.

[11] Ahamed Yasar Z, Dinesh Kumar Rand else," Border Surveillance System using Computer Vision", 2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS), 978-1-7281-5197-7/20/2020 IEEE.

[12] Kumar Shukla, Ratnesh, and Arvind Kumar Tiwari. "Comparative analysis of machine learning based approaches for face detection and recognition." *Journal of Information Technology Management* 13.1 (2021): 1-21.

[13] Lu, Yuhang, Luca Barras, and Touradj Ebrahimi. "A novel framework for assessment of deep face recognition systems in realistic conditions." *2022 10th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2022.

[14] Zhang, Huaizhong, et al. "A novel infrared video surveillance system using deep learning based techniques." *Multimedia tools and applications* 77 (2018): 26657-26676.

[15] Harikrishnan, J., et al. "Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision." *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)*. IEEE, 2019.

[16] Gupta, Aakansha, and Rahul Katarya. "Social media based surveillance systems for healthcare using machine learning: a systematic review." *Journal of biomedical informatics* 108 (2020): 103500.

[17] Patel, Kshitij, and Meet Patel. "Smart surveillance system using deep learning and RaspberryPi." *2021 8th International Conference on Smart Computing and Communications (ICSCC)*. IEEE, 2021.

[18] Chen, Jianguo, et al. "Distributed deep learning model for intelligent video surveillance systems with edge computing." *IEEE Transactions on Industrial Informatics* (2019).

[19] Zhao, Rui, et al. "Deep learning and its applications to machine health monitoring." *Mechanical Systems and Signal Processing* 115 (2019): 213-237.

[20] Wagner, Maximilian, et al. "A super low-cost bioprinter based on DVD-drive components and a raspberry pi as controller." *Bioprinting* 23 (2021): e00142.

[21] Lamine, Hayet, and Hafedh Abid. "Remote control of a domestic equipment from an Android application based on Raspberry pi card." *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. IEEE, 2014.

[22] Nayak, Maya, and Prasannajit Dash. "Smart surveillance monitoring system using Raspberry Pi and PIR sensor." *Statistics* (2014).

[23] Maragatham, T., P. Balasubramanie, and M. Vivekanandhan. "IoT Based Home Automation System using Raspberry Pi 4." *IOP Conference Series: Materials Science and Engineering*. Vol. 1055. No. 1. IOP Publishing, 2021.

[24] Gunawan, Teddy Surya, et al. "Development of face recognition on raspberry pi for security enhancement of smart home system." *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)* 5.4 (2017): 317-325.

[25] Chellappa, Rama, Pawan Sinha, and P. Jonathon Phillips. "Face recognition by computers and humans." *Computer* 43.2 (2010): 46-55.

[26] Wechsler, Harry, et al., eds. *Face recognition: From theory to applications*. Vol. 163. Springer Science & Business Media, 2012.

[27] Lin, Kwan-Ho, Kin-Man Lam, and Wan-Chi Siu. "Spatially eigen-weighted Hausdorff distances for human face recognition." *Pattern Recognition* 36.8 (2003): 1827-1834.

[28] Gao, Hongbo, et al. "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment." *IEEE Transactions on Industrial Informatics* 14.9 (2018): 4224-4231.

[29] Wang, Cheng, et al. "NormalNet: A voxel-based CNN for 3D object classification and retrieval." *Neurocomputing* 323 (2019): 139-147.

[30] Miron, Casian, Alexandru Pasarica, and Radu Timofte. "Efficient cnn architecture for multi-modal aerial view object classification." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

[31] Akcay, Samet, et al. "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery." *IEEE transactions on information forensics and security* 13.9 (2018): 2203-2215.

[32] Guérin, Joris, et al. "CNN features are also great at unsupervised classification." *arXiv preprint arXiv:1707.01700* (2017).

[33] Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.

[34] Kausar, Asia, et al. "Pure-cnn: A framework for fruit images classification." *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018.

[35] Wang, Yi-Qing. "An analysis of the Viola-Jones face detection algorithm." *Image Processing On Line* 4 (2014): 128-148.

[36] Dabhi, Mehul K., and Bhavna K. Pancholi. "Face detection system based on Viola-Jones algorithm." *International Journal of Science and Research (IJSR)* 5.4 (2016): 62-64.

[37] Hefenbrock, Daniel, et al. "Accelerating Viola-Jones face detection to FPGA-level using GPUs." *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2010.

[38] Chaudhari, Monali Nitin, et al. "Face detection using viola jones algorithm and neural networks." *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018.

[39] Tavallali, Pooya, Mehran Yazdi, and Mohammad Reza Khosravi. "An efficient training procedure for viola-jones face detector." *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2017.

[40] Kumar, Pranav, S. L. Happy, and Aurobinda Routray. "A real-time robust facial expression recognition system using HOG features." *2016 International Conference on Computing, Analytics and Security Trends (CAST)*. IEEE, 2016.

[41] Dadi, Harihara Santosh, and GK Mohan Pillutla. "Improved face recognition rate using HOG features and SVM classifier." *IOSR Journal of Electronics and Communication Engineering* 11.4 (2016): 34-44.

[42] Mizuno, Kosuke, et al. "Architectural study of HOG feature extraction processor for real-time object detection." *2012 IEEE Workshop on Signal Processing Systems*. IEEE, 2012.

[43] Kobayashi, Takumi. "BFO meets HOG: feature extraction based on histograms of oriented pdf gradients for image classification." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013.

[44] Dahmane, Mohamed, and Jean Meunier. "Emotion recognition using dynamic grid-based HoG features." *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE, 2011.

[45] Yue, Shihong, Ping Li, and Peiyi Hao. "SVM classification: Its contents and challenges." *Applied Mathematics-A Journal of Chinese Universities* 18 (2003): 332-342.

[46] Chandra, Mayank Arya, and S. S. Bedi. "Survey on SVM and their application in image classification." *International Journal of Information Technology* 13 (2021): 1-11.

[47] Mavroforakis, Michael E., and Sergios Theodoridis. "A geometric approach to support vector machine (SVM) classification." *IEEE transactions on neural networks* 17.3 (2006): 671-682.

[48] Thanh, Dang Ngoc Hoang, and Serdar Engínoğlu. "An iterative mean filter for image denoising." *IEEE Access* 7 (2019): 167847-167859.

[49] Zhu, Rong, and Yong Wang. "Application of Improved Median Filter on Image Processing." *J. Comput.* 7.4 (2012): 838-841.

[50] Cabello, Frank, et al. "Implementation of a fixed-point 2D Gaussian Filter for Image Processing based on FPGA." *2015 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE, 2015.

[51] Ramakrishnan, S., ed. *Face Recognition: Semisupervised Classification, Subspace Projection and Evaluation Methods*. BoD–Books on Demand, 2016.

[52] Jamborsalamati, Pouya, et al. "A framework for evaluation of power grid resilience case study: 2016 South Australian blackout." *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2018.

[53] Zhang, Mengli, et al. "AsU-OSum: Aspect-augmented unsupervised opinion summarization." *Information Processing & Management* 60.1 (2023): 103138.

[54] Saxena, Abhinav, et al. "On applying the prognostic performance metrics." *Annual Conference of the PHM Society*. Vol. 1. No. 1. 2009.

[55] Islam, Md Adnanul, and Md Saddam Hossain Mukta. "A comprehensive understanding of popular machine translation evaluation metrics." *International Journal of Computational Science and Engineering* 25.5 (2022): 467-478.

[56] https://www.kaggle.com/datasets/vaspatel/face-recognition-dataset

CURRICULUM VITAE

Rusul Naseer ALLAMI

Education

- **MSc. In Electrical and Electronics** **2022-2023**
  University of Gaziantep, Engineering Faculty, Department of Electrical and Electronics Engineering.
- **BSc. In Computer Engineering** **2017-2018**
  AL-RAFIDAIN UNIVERSITY COLLEGE – Department of computer Engineering