Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Babylon

College of Science for Women

Department of Computer Science

Person Identification Based on Deep Fusion of Iris Features and Machine Learning Approaches

A Thesis

Submitted to the council of College of Science for woman,

University of Babylon in Partial Fulfillment of the Requirement

For Degree of Master of Science in Computer Science

Submitted by

Israa Adil Hassan

Supervised by

Dr. Suhad A. Ali

Dr. Hadab Khalid Obayes

2022 A.D. **1444 A.H.**

بسم الله الرحمن الرحيم

(قالوا سبحانك لا علم لنا الا ما علمتنا انك انت العليم الحكيم)

صدق الله العلي العظيم

سورة البقرة – الاية(32)

Dedication

To

My Twelfth Imam Al-Hujjah Ibn Al-Hasan, who will fill the earth with equity and justice after it has been filled with injustice and oppression.

To

those who sacrificed themselves in defense of their land,

Our righteous martyrs,

martyr leaders.

To

the soul of my father who always wanted me to complete my studies, but he passed away before he witnesses this day.

To

my family,

I dedicate this humble effort.

Supervisors Certification

we certify that this thesis entitled "***Person Identification based on Deep Fusion of Iris Features and Machine Learning Approaches***" was done by (*Israa Adil Hassan*) under our supervision.

Signature:

Name: ***Prof. Dr. Suhad Ahmed Ali***

Date: / / 2022

Address: University of Babylon/College of Science for Women

Signature:

Name: ***Dr. Hadab Khalid Obayes***

Date: / / 2022

Address: College of Education for Humanities Studies, University of Babylon

The Head of the Department Certification

*In view of the available recommendations, I forward the dissertation entitled "***Person Identification Based on Deep Fusion of Iris Features and Machine Learning Approaches***" for examining committee.*

Signature:

Name:

Date: / / 2022

Address: University of Babylon/College of Science for Women

Acknowledgments

All thanks and praise to Allah, the Lord of the world, who gave me courage and enabled me to achieve this work.

My thanks and gratitude go to my supervisors ***Dr. Suhad Ahmed Ali*** and ***Dr. Hadab Khalid Obayes*** for the support and guidance they have given me and the effort and time to complete this research.

Thanks, and gratitude to all my professors and all the staff of the Department of Computer Sciences \ College of Sciences for Women \University of Babylon for their help.

All Thanks and Gratitude to all my family and friends for their support and encouragement.

Israa Adil ... (2022)

Abstract

A biometric system provides automatic identification of an individual based on a unique feature or characteristic possessed by the individual. Iris recognition is one of the most representative identification technologies in biometric recognition, which is widely used in various fields. These systems depend on iris texture features, which are characterized by richness, randomness, and uniqueness, in addition to stability, and this makes iris recognition systems more accurate and reliable.

Irregularities in iris images (like poor quality, nonlinearly deformed iris images) make the recognition task harder and more challenging. Also, the inherent noise in iris images causes significant degradation in cognition efficiency. All these problems open challenges in iris recognition topic. Hence, it is important to develop an effective and accurate iris identification system that enhances the identification accuracy. To address these needs, in this thesis, iris identification system based on finding a set of iris features that enhance the system accuracy is investigated.

In recent years, deep learning has achieved high performance in many computer vision tasks, such as image classification, semantic segmentation, and object detection. Therefore, deep learning techniques have been relied upon to extract iris texture features. To extract the features, the deep learning technique represented by Convolutional Neural Network (CNN) is used. The CNN Network is designed to consist of several layers to extract the features.

Through the phases of training and testing, the input eye image is passed through a sequence of main processing stages (i.e., iris Segmentation, normalization and features extraction). Segmentation is the process of separating the iris region from the rest of the eye parts. First, the iris region is localized by first finding the boundary of pupil by using a set of image processing operations.

Second, a method for identification of the outer boundary of iris region is applied. To bypass the problems, arise due to the normal change in pupil size the circular iris is resampled & mapped to be rectangular. The iris is located between the pupil and the white sclera, so it has an annular shape and an unstable size, so another process is needed to turn it into a fixed-dimensional rectangle whose features are easy to extract. This process is called Normalization.

To show the impact of extracted features by proposed CNN model, several methods were used to identify the person identity. The first method of classification was using the artificial neural network (ANN) represented by the fully connected layer in CNN model and SoftMax classifier. The second one includes classification by using minimum Euclidean Distance. In this method, the extracted features in the training phase are used to build a template of each person based on applying average fusion method on training samples of each person which is saved in train database. The third classification method is to use a Support Vector Machine (SVM).

The developed system was tested on two databases (CASIA-1 and CASIA-4). The accuracy results for first method were 98.89, 98.5 for the CASIA-V1 and CASIA-V4 respectively. The accuracy results for the second method were 96.29%, and 95.83 for CASIA-V1 and CASIA-V4 respectively. The accuracy results for third method were 99.54%, and 98.83% for CASIA-V1 and CASIA-V4 respectively.

List of Content

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Meaning
Adam	*Adaptive Moment Estimation*
AI	*Artificial Intelligent*
ANN	*Artificial Neural Network*
BGD	*Batch Gradient Descent*
BN	*Batch Normalization*
CHT	*Circular Hough Transform*
CNN	*Convolutional Neural Network*
Conv	*Convolutional*
DL	*Deep Learning*
DNA	*Deoxyribonucleic acid*
DNN	*Deep Neural Network*
DRDL	***D**ynamic **R**outing-based algorithm (depend on **D**irection &**L**ength information of vectors)*
ERR	*Energy Efficiency Ratio.*
FC	*Fully Connected*
FN	*False Negative*
FP	*False Positive*
FRR	*False Rejection Rate*
GPU	*Graphical Processing Unit*
HT	*Hough Transform*
LoG	*Laplacian of Gaussian*
M-BGD	*Mini- Batch Gradient Descent*
ML	*Machine Learning*
MLP	*Multi-Layer Perceptron*
MMSD	*Minimum shifted and Masked Distance*
PCA	*Principle Component Analysis*
ReLU	*Rectified Linear Unit*
RNN	*Recurrent Neural Network*
ROI	*Region Of Interest*
S	*SoftMax*
SE	*Structure Element*
SGD	*Stochastic Gradient Descent*
SGDM	*Stochastic Gradient Descent with Momentum*
SIFT	*Scale Invariant Feature Transforms*
SVM	*Support Vector Machine*
T	*Threshold*
TN	*True Negative*
TP	*True Positive*

List of Symbols

Symbol	***Meaning***
θ	*Theta (angle magnitude)*
δ	*The change amount*
γ	*Gamma Correction*
σ	*Standard Deviation of the Distribution*
$\oplus$	*Morphological Dilation Operation*
$\ominus$	*Morphological Erosion Operation*
$\circ$	*Opening (Erosion &Dilation)*
$\bullet$	*Closing (Dilation & Erosion)*
$\mathit{\Theta}$	*Hadamard Product*
η	*Learning Rate*

List of Algorithms

Chapter One

General Introduction

Chapter One

General Introduction

1.1 Introduction

Nowadays, the great interest in biometric recognition systems can be justified due to the increased demand for security, additionally, it's having some advantages: cannot be stolen, lost, or forgotten. However, to be reliable, biometric traits should be unique and persistent over time. A biometric feature can be defined as a physiological (face, fingerprints, iris, etc.) or behavioral (gait, voice, signature, etc.) attribute of a human being that can discriminate one individual from another [1].

Iris recognition is considered one of the important methods of effective personal identification because it has many characteristics [2]. The characteristics of the iris can be summarized: Firstly, the iris trait represents the annular region of the eye lying between the black pupil and the white sclera; this makes it completely protected from varied environmental conditions [2]. Secondly, it is believed that the iris texture provides a very high degree of uniqueness and randomness, so it is very unlikely for any two iris patterns to be the same, even irises from identical twins, or from the right and left eyes of an individual. This complexity in iris patterns is due to the distinctiveness and richness of the texture details within the iris region, including rings, ridges, crypts, furrows, freckles, and zigzag patterns[3]. Thirdly, the iris trait provides a high degree of stability during a person's lifetime from one year of age until death. Finally, it is considered the most secure biometric trait against fraudulent methods and spoofing attacks[4].

Usually, the input images of the eyes are in un-constrained conditions. Meaning the algorithm should detect and identify the iris area. This operation is considered complicated due to the noise and variation of the iris location[5]. Thus, iris location should be identified and detected first in order to process it later. The features of the iris region are extracted after being selected for matching operations. A range of algorithms is provided for feature extraction, including traditional techniques like 2D Gabor filters, Wavelet transforms, and Laplacian of Gaussian filters as well as newer techniques like Convolutional Neural Network (CNN) [6]. Extracting effective features is the major important stage in a lot of object recognition and computer vision tasks. Therefore, several researchers have focused on designing robust features for a variety of image classification tasks. Nowadays, much attention is given to feature learning algorithms and Convolutional Neural Networks (CNN)[7][8]. Here, the meaning of the concept of feature is any distinctive aspect or characteristic which is used to solve a computational task related to a certain application. The combination of *n* features can be represented as an *n*-dimensional vector, called a feature vector. The quality of a feature vector is dependent on its ability to discriminate image samples from different classes[9].

1.2 Related Works

Most of research in iris recognition system field deals with one of the problems that is found in the relevant steps of iris recognition system (i.e. iris segmentation, iris features extraction, and pattern matching) in order to get accurate results. This section lists two sub-sections of research, the first sub-section contains some research that includes some localization methods, in the second section, several recent systems of iris recognition have been reviewed. Also, these sub-sections include some of the latest studies about deep learning and its role in the identification system.

1.2.1 Related Works (Iris Segmentation)

The most effective and modern approaches in detecting the iris could be grouped into two general approaches. The first approach involves using two typical algorithms were proposed by Daugman (integro-differential operators) and Wildes (Hough Transform) [10]. The second approach involves using Deep Learning techniques.

A. Approaches Using Daugman and Wildes Methods and other methods

In 2018, Kennedy and others proposed a method for iris segmentation. It entails switching from the integro-differential operator approach (John Daugman's model) to the Hough transform (Wilde's model) as the segmentation strategy for this implementation. This study analyzed the two segmentation approaches in-depth to determine which is superior for recognition based on wavelet packet decomposition. The integro-differential technique to segmentation was found to be 91.39 percent accurate, whereas the Hough Transform approach was found to be 93.06 percent accurate[11].

In 2021, Ahmed and others proposed a method for iris segmentation. In this method for pupil boundary detection, processes such as morphological filtering and two-direction scanning were applied. The Wildes approach (Hough Transform) is adjusted for limbic border identification by limiting the Canny edge detector and Hough transform processes to a tiny Region of Interest (ROI) not exceeding 20% of the picture area. This method was tested on several databases, including CASIA-V1and V4, and the accuracy rate was 96.48 - 95.1 respectively [12].

In 2022, Tariq M. and others presented a compound method to perform the iris segmentation based on several techniques such as LoG Filter (Laplacian of Gaussian), region growing (One of the ways to segment images), and zero-crossings of the LoG filter. In this suggested method, to detect the pupil region used LoG filter with region growing, and used zero-crossings of the LoG filter to correctly identify the boundaries of the inner and outer circular iris. This method has been tested on several public databases including CASIA-V1 and CASIA-

V3. The segmentation accuracy of the proposed method was good and outperformed many methods [13].

B. Approaches Using Deep Learning Methods

In 2018, Lozej and others proposed a model based on U-Net to perform iris segmentation. The architecture of U-Net is known in the medical image processing field due to its high performance on a relatively small dataset. It uses the encoder-decoder design. The encoder is performing classical CNN operations. In this work, the VGG model is used for the encoder. The decoder up-samples the lower layer's feature maps while also concatenating and cropping the encoder part's output of the same depth. The training technique employs adaptive moment estimate (Adam) and binary cross-entropy. CASIA database is used with 160 images for the training phase, and 40 images for the testing phase. Accuracy ranged between 96% - 97% based on network depth and Batch Normalization[14].

Another U-Net based work is proposed in 2019 by Zhang and others. However, they extracted more global features using dilated convolution rather than original convolution to better process the details of images. In fully dilated convolution (FD-UNet), the convolutional mask has zero values inside of it (i.e., avoid some parts of the original image). This will lead to more receptive field information without increasing the complexity of the algorithm and losing the information. This method was tested on several databases, including CASIA-Iris V4 (interval), and the accuracy rate was 97,36% [15].

In 2019, Yung-Hui Li and others presented a method composed of learning and edge detection algorithms for iris segmentation. The bounding box is found by Faster R-CNN that is built of six layers; the region of the pupil is detected by using a Gaussian mixture model. On the CASIA-Iris-Thousand database, experimental findings for this proposed technique obtained 95.49 % accuracy [16].

Table (1.1) presents the summary of all listed works related to iris segmentation stage.

Table (1.1): Summary of the Literature review for Iris Segmentation Methods

References	Year	Method	Dataset	Accuracy%
[11]	*2018*	*Integro-differential operator*	*CASIA-Iris*	*91.39*
		Hough Transform		*93.06*
[12]	*2021*	*Morphological filter &two direction scanning*	*CASIA-V1*	*96.48*
			CASIA-V4	*95.1*
[13]	*2022*	*Laplacian of Gaussian (LOG) filter*	*CASIA-V1*	*100*
			CASIA-V3	*99.55*
[14]	*2018*	*U-Net*	*CASIA-V1*	*ranged between 96% - 97% based on network depth and BN*
[15]	*2019*	*Fully dilated convolution combining U-Net (FD-UNet)*	*CASIA-V4*	*97.36*
[16]	*2019*	*Faster R-CNN*	*CASIA-V4*	*95.49*

1.2.2 Related Work (Iris Identification Systems)

In 2017, Kien Nguyen and others presented Iris Recognition System with Off-the-Shelf CNN Features. This work consists of several steps: Segmentation (used Integro-Deffrentioal Operator), Normalization (used Rubber Sheet Model), Feature Extraction ((using Off-the-Shelf CNNs), where a number of pre-trained CNN architectures has been applied such as, VGG-Net, Dense-Net, and RES-Net& Inception), and finally classification (using Support Vector Machine (SVM)). This system was tested on several databases including CASIA-Iris-Thousand. Where VGG-Net, ,Dense-Net, and RES-Net& Inception networks achieved results with accuracy up to(93.1% with 9 layers),(98.8% with 5 layers), and (98.3% (10 layers), 98.5% (12 layers)) respectively[17].

In 2018, Maram.GAlaslani and Lamiaa A. Elrefaei1 evaluated the extracted learned features from a pre-trained Convolutional Neural Network (Alex-Net Model) followed by a multi-class Support Vector Machine (SVM) algorithm to perform classification. The performance of the proposed system is investigated when extracting features from the segmented iris image and from the normalized iris image. The proposed iris recognition system is tested on several datasets: CASIA-Iris-V1, CASIA-Iris-thousand and, CASIA-Iris- V3 Interval. 60 classes were used from each dataset and the results were 98.3%, 98%, and 89% respectively[18].

In 2018, Yuslena Sari and others proposed a new feature extraction based on iris texture patterns with Principal Component Analysis (PCA). PCA is used to store computing process in classification process. The focus of this research was to compare the accuracy result of classification methods of distance measurement such as Euclidean Distance, City Block Distance, Chebyshev Distance, and other. Accuracy test shows that PCA can be used in various classification methods that use distance measurement. This method was tested on CASIA- V4 Database and the results of accuracy was 85.0 for the Euclidean Distance measure. [19]

In 2019, T.Zhao and others proposed iris recognition system based on Deep Learning Techniques. After pre-processing the iris images to obtain the normalized images, Capsule network deep neural architecture is applied for iris identification purpose. Three cutting-edge pre-trained models are introduced: VGG16, InceptionV3, and ResNet50. The three networks are divided into a series of subnetwork architectures based on the number of primary constituent blocks. Instead of a single convolutional layer in the capsule network, they are used as the convolutional component to extract primary features. Three models have been proposed InceptionV3_5block+DRDL, Iris-Dence+DRDL, and VGG16_4block+DRDL. The term DRDL stands for **d**ynamic **r**outing-based algorithm, which depends on the **d**irection and **l**ength information of the vectors, so it is called DRDL. The proposed method was tested on the CASIA-V4 dataset. VGG16_4block+DRDL model has achieved the highest accuracy for this dataset, reaching93.87%[20].

In 2019, Maram.G Alaslni1 and Lamiaa A. Elrefaei proposed iris recognition system based on Transfer Learning with Convolutional Neural Network. This work, like any iris recognition system, consists of several stages, from the initial processing of the iris images to the extraction of features using the principle of transfer learning with CNNs. The proposed system is implemented by fine-tuning a pre-trained Convolutional Neural Network (VGG-16) for feature extraction and classification. The performance of the iris recognition system is tested on CASIA-Iris-V1,CASIA-Iris Thousand, and CASIA-Iris Interval and the results were 98.3%, 95%, and 91.6% respectively[6].

In 2020, SaiyedUmer and others presented a method for iris recognition that consists of a number of stages: preprocessing, feature extraction, and classification. In image preprocessing an annular iris, the portion is segmented out from an eyeball image and then transformed into a fixed-sized image region. The parameters of iris localization have been used to extract the local periocular region. For features extraction and classification tasks VGG16, ResNet50, and Inception-v3 CNN architectures have been employed. The proposed method applied on several databases including CASIA-Iris-Distance and the accuracy results ranged between 96-98% dependence on the CNN architecture used[21].

In 2021, K.Kumar and others proposed a system for iris recognition based on several techniques such as Circular Hough transform for iris segmentation, Doughman Rubber Sheet Model for performing Normalization to obtain images in a unified form, and then used CNN(Mini-VGG) which has not been educated before. This Net Consists of two sets of Convolutional and ReLU layers, Max pooling, Collection of fully connected layers. This work used CASIA-Iris V1(Dataset consisting of 756 iris images from 108 folders having 7 samples each) and split it into 80 % for the training phase and 20% for the testing phase to validate the model performance on unseen data. Experimental results were accuracy, precision, recall is 0.98%,0,99% and 0.99% respectively[22].

In 2021, S.Sujana and VSK Reddy proposed an iris recognition system based on two phases: pre-processing is the first phase that includes iris segmentation and Normalization, that used Circular Hough Transform with Canny Edge Detection to obtain the segmented images and Doughman Rubber Sheet Model for Normalization. The second phase include sex tracing the features from the normalized images by using deep learning techniques (CNN Architecture). The performance of this architecture depends on hyper-parameters tuning, optimizer used, and learning rate schedules. This system has been tested on CASIA-Iris V1(where the accuracy was 95.4%)[23].

In 2021 A. N. Hashim, R. R. Al-Khalidy suggested a developed system for iris identification. This system is based on the fusion of scale-invariant feature transforms (SIFT) and local binary patterns of features extraction. This work applied in several steps include: convert images to grayscale, perform iris segmentation by using Circular Hough transform, for Normalization used Daugman's rubber sheet models, apply the histogram equalization to enhance the iris region, and finally, extract the features by using the scale-invariant feature transformation (SIFT) and local binary pattern. SIFT Algorithm consist of four stages: (Detection of Scale Space Extreme, Accurate Localization Extreme Points, Orientation and KeyPoint descriptor).This system has been tested on CASIA-Iris V4 Dataset ((30 class) every class has 20 samples, with a total of 600 images) and was the accuracy 98.67% for left eyes and 96.66% for right eyes [24].

In 2021, Kai Yang and others proposed a network architecture based on the dual spatial attention mechanism for iris recognition, called DualSANet. This work includes several steps: Image Pre-processing to obtain normalized images, Network Architecture to extract dual spatially corresponding iris feature representations for iris recognition. This net is performed by encoder-decoder structure. CASIA V4 Thousand and Distance Dataset is among the databases to which this architecture has been applied, and the results show that this method achieved superior performance compared with the existing methods. (FRR and EER for CASIA-IrisV4-Thousand was 0.31%, 0.27% and for CASIA-IrisV4-Distance was 10.67%, 3.23% respectively[25].

In 2022, Suzwani Ismail and others experimented with 17 pre-trained CNN networks such as DarkNet-19, SqueeseNet, Alex-Net, Google Net, ... etc.) with a different number of epochs in each experiment, starting from 4 epoch to 20 on a dataset CASIA-Iris Lamp. A multi-class Support Vector Machine (SVM) algorithm used to perform classification task. In this work utilized different epoch number on the model until a good accuracy is achieved with different types of layers in CNN in order to identify which type of layers could extract good features. Simulation results of 92% by DarkNet-19 on CASIA-Iris-Lamp dataset reveal the effectiveness of deep learning algorithms on extracting irregular features of deformed iris[26].

Table (1.2) presents the summary of all listed works related to iris recognition system.

Table (1.2) Summary of Literature review for Iris Recognition System

References	Year	Feature Extraction	Classification	Dataset	Accuracy%
[17]	2017	VGG-Net	SVM Classifier	CASIA-Iris Thousand	93.1
		Dence-Net			98.8
		Res-Net &Inception			98.0
[18]	2018	Alex-Net Model	SVM Classifier	CASIA-V1	98.3
				CASIA-Thousand	98
				CASIA-V3	89
[19]	2018	Principle Component Analysis (PCA)	Similarity Distance Measures	CASIA -V4	85.0
[20]	2019	VGG16-4block+DRDL	SoftMax Classifier	CASIA-Iris-V4	93.87
[6]	2019	Pre-trained VGG16	SoftMax Classifier in Pre-trained Network	CASIA-Iris V1	98
				CASIA-V4 Thousand	95
				CASIA-Iris Interval	91.6
[21]	2020	VGG16	SoftMax Classifier in Pre-trained Network	CASIA-Iris-Distance	96.79
		ResNet 50			97.68
		Inception v3			98.69
		Inception- ResNet			98.43
[22]	2021	Mini-VGG16	SoftMax Classifier	CASIA-Iris V1	98
[23]	2021	Proposed CNN Architecture	SoftMax Classifier	CASIA-Iris V1	95.4
[24]	2021	SIFT and Local Binary pattern	City Block Distance	CASIA-Iris-V4	96.66 98.67
[25]	2021	DualSANet Model	Minimum shifted and Masked Distance (MMSD)	CASIA-V4 Thousand CASIA-V4 Distance	---
[26]	2022	Darknet19 Model	SVM Classifier	CASIA-Iris-Lamp	92

1.3 Problem Statement

The limitation of traditional iris recognition systems to process iris images captured in unconstraint environments is challenging. Automatic iris recognition has to face unpredictable variations of iris images in real-world applications. For example, the most challenging problems are related to the severe noise effects that are inherent to these unconstrained iris recognition systems. The typical sources of noise include motion blur, lighting, the distance between the person and the camera, and specular reflection, among others. The most difficult challenge is to develop an accurate and reliable iris recognition system for use in security applications, which is the key reason for this thesis. Finding robust iris features representation for a variety of images can enhance the recognition accuracy.

1.4 Aims of Thesis

In order to achieve the aim of high accurate person identification, thus sub aims are required to be accomplished including:

1. Iris localization is an important step in iris recognition system, thus design an algorithm to segment the region of interest (Iris region in the image of eye) based on a combination of image processing technique and Hough transform.
2. The main objective of this thesis is to overcome the poor robustness of the classical hand-crafted feature extraction method in the iris recognition system. Design and implement an iris identification system using Deep Learning for iris images to discover a feature representation scheme that is primarily data-driven.
3. By automatically learning the feature representation from the iris data, an optimal representation scheme can potentially be deduced by building iris

template based on features fusion, leading to high recognition results for the iris recognition task.

1.5 Thesis Outline

The remaining part of the thesis is organized as the following:

- Chapter two: "Theoretical Background"

This chapter will provide a background of Iris Recognition Systems, Segmentation and Normalization techniques used in the proposed system, and an introduction about Machine Learning and Deep Learning (CNN Architectures).

- Chapter three: "Proposed Iris Identification System"

A thorough explanation of the main stages for building the Iris Recognition Systems, Convolutional Neural Network (CNN), and the selected algorithm of the Deep Learning to tackle our problem is presented. This chapter contains all the principles of Iris Recognition Methods, Convolutional Neural Networks, besides the related researches and published works and codes that may be useful and insightful for our understanding of how this technique is working. Moreover, the design of our work and algorithms are explained.

- Chapter four: "Experimental Results and Evaluation"

The experimental results of the proposed system will be presented in this chapter.

- Chapter five: “Conclusions and Future Works”

This chapter shows the major conclusions gotten from the results of the proposed system in addition to the suggestions of some future works.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1 Introduction

This chapter includes the different approaches and mechanisms that have been implemented in the iris recognition systems. And also includes a detailed explanation of the image processing techniques that used in these systems.

The basic stages of building the iris recognition systems are Image Acquisition, Iris Segmentation, Iris Normalization, Features Extraction, and the classification. These stages will be clarified during the chapter in some details and an explanation of the techniques used, both manual and related to machine learning and deep learning.

2.2 Biometric Systems

Biometrics is the most promising system for identifying a user, where it is associated with uniquely human characteristics. Figure (2.1) shows types of biometric identifiers. Biometric authentication can be preferred over many traditional methods, such as smart cards and passwords because biometrics makes information difficult to steal. Physiological traits such as fingerprints, DNA, facial recognition, iris, and so on, and behavioral characteristics such as voice, gait, signature, and so on, are the most frequent biometric identifiers [23].

Some of the principal properties of a biometric system are the following [27]:

1- **Robustness (Stability)**: the biometric characteristics have to be highly unchangeable over a period of time.

2- **Individuality**: biometric characteristics should not be identical in two individuals.

3- **Availability**: every person should be possessed biometric characteristics.

4- **Accessibility**: the degree that which the users are ready to agree to use a specific biometrical feature (characteristic) in their everyday routine.

5- **Unforgettable**: biometric characteristics are unforgettable because they depend on the human or behavioral characteristics of each person.

6- **Not subject to theft or guesswork**: biometric characteristics cannot be stolen or guessed as in a smart card or password because they are linked to human and behavioral characteristics specific to each person.

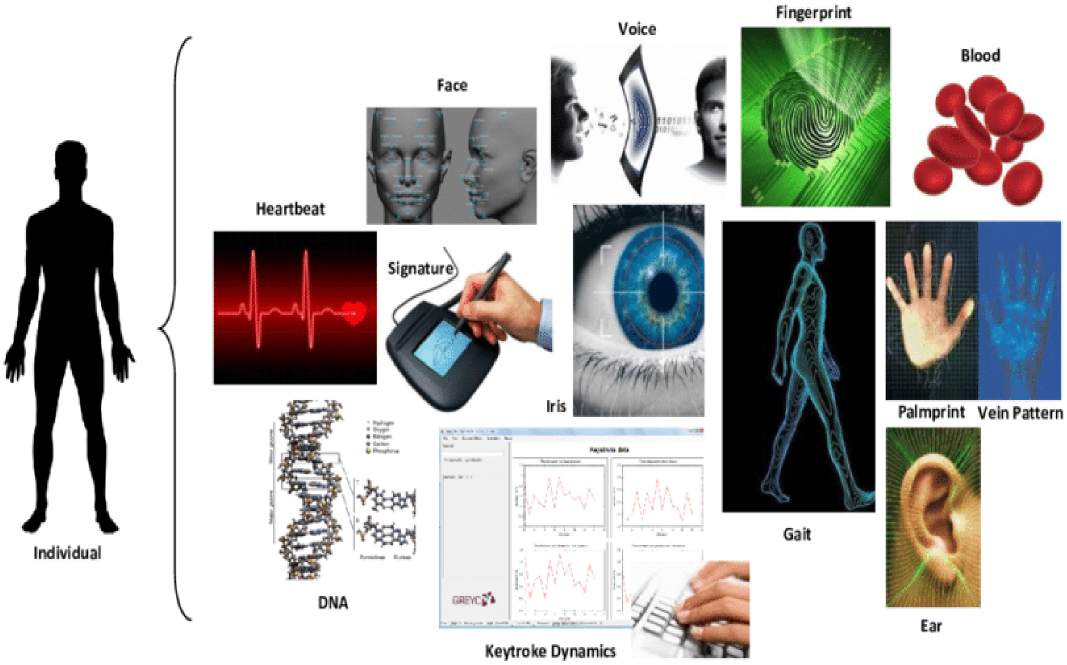


Figure (2.1): Types of Biometric Identifiers [28]

Various Biometric Identification Techniques in use today:

A. **Physiological Biometric Features:**

- **Face Recognition**: Facial recognition systems are a technology that can identify a person or verify their identity through a digital image, where facial features are extracted [29].

- **Fingerprint Recognition**: The ridges and valleys on the surface tips of a human finger is used to identify a person [30].

- **Palmprint Recognition**: It is the extraction of the features of the palmprint to identify the person or verify his identity [31].

- **Eyes**: That includes **Iris Recognition** and **Retina Recognition**. Iris Recognition uses the ***Iris*** features to identify a person whereas Retina Recognition uses the ***Veins patterns*** in the back of the eye to recognize people.

- **Ear Recognition**: Using the shape of a person's ear to identify them.

- **DNA Matching**: The identification of a person based on the analysis of DNA segments [30].

A. Behavioral Biometric Features [30]**:**

- **Gait Recognition**: Using a person's walking style or gait to determine their identity.

-**Signature Recognition**: The study of a person's handwriting style, especially a signature, to verify their identity.

- **Voice/Speech Recognition**: The use of a person's voice to determine their identity.

- **Keystroke Dynamics Recognition**: They are systems that have the ability to authenticate the user by studying typing patterns[32].

2.3 Parts of the Human Eye

Before getting into the details of the iris recognition systems, let us mention a brief overview of the main parts of the human eye as shown in Figure (2.2):

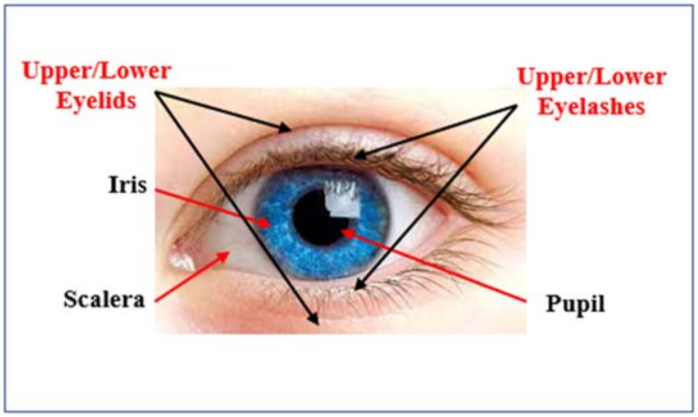


Figure (2.2): Parts of a Human Eye

- **Upper/Lower Eyelids:** A slender skin crease that extends out and secures the human eye.
- **Upper/Lower Eyelashes:** These are the hairs that go along the edge of the eyelid to protect it.
- **Sclera:** The white covering of the eyeball on the outside.
- **Pupil:** It is a gap in the iris of the eye’s focal point that allows light to strike the retina (Retina is a thin layer of tissue that lines the back of the eye on the inside and is more sensitive to light than other eye tissues, it is convert the light into neural signals and send these signals on to the brain for visual recognition).
- **Iris:** It is a thin, spherical structure in the eye that regulates the width and size of the pupil and, as a result, the amount of light that reaches the retina And it is represent the colored part at the front of the eye that contains the pupil in the center [30].

2.4 Iris Recognition Systems

The automated method of detecting individuals based on their iris patterns is known as iris recognition. The iris stroma's seemingly random character distinguishes it as a biometric recognition trigger[17]. These systems ranked high among the biometric systems due to the accuracy and high efficiency that they are characterized by [24]. Where the iris is characterized by continuity, uniqueness, and stability [33], therefore, it is used in high-security environments such as airports and harbors to control borders.

Iris recognition system has major stages of construction as shown in Figure (2.3). For the iris recognition system, several algorithms have been employed, and we will review the most essential algorithms in each iris recognition stage.

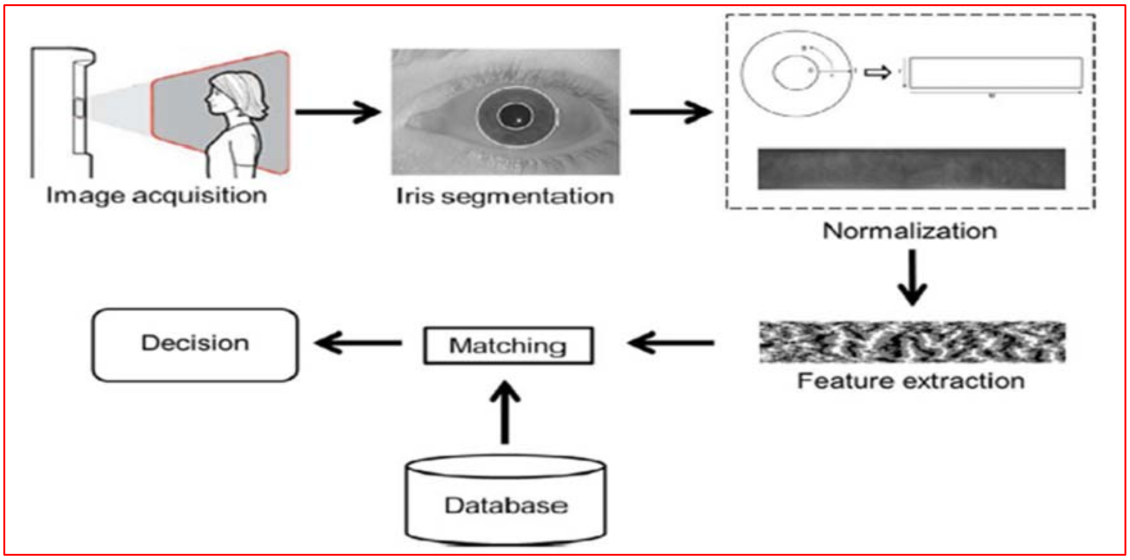


Figure (2.3): The main stages of the Iris Recognition System [34]

2.4.1 Image Acquisition

Image Acquisition is the first step in the iris recognition system. It takes the iris images by using special cameras developed for this purpose in different environments[6]. The databases differ according to the spectrum in which they were captured, such as the near-infrared spectrum, visible light, or other. It is possible to take images of the iris under a specific environment and work on them through databases available free of charge via the internet, as about 65% of databases were taken by infrared spectrum and the rest in the visible spectrum[35]. From databases that are freely available on the internet: CASIA-Iris Dataset, IITD-Iris Dataset, UBIRIS-Iris Dataset …etc.

2.4.2 Iris Segmentation

Iris segmentation is the process of detecting the location of the iris area in the eye image. So, it will be used later to identify the identity of the person with that eye. It involves defining the inner and outer borders of the iris, which is crucial for the accuracy of iris recognition systems [36].Additionally, this stage of segmentation also allows for the normalization of the iris region and the extraction of discriminative features from well-aligned iris samples [37].

This stage, along with the post-normalization stage, is a pre-processing of the iris image before the features are extracted and before they are applied, the image is enhanced so that the segmentation results are correct and accurate. Image enhancement methods are used to eliminate noise such as median filter, Gaussian filter, and others, in order to get a clearer iris image.

The most effective approaches in detecting the iris could be grouped into two general approaches. The first approach involves using algorithm proposed by Daugman called Integro-differential operators and the second approach depends on the Hough Transform that proposed by Richard Duda and Peter Hart[30] .

I. Daugman’s Integro-Differential Operators

In 1993, John Daugman presented the first successful iris segmentation algorithm which uses Integro Differential operator, so it is called Daugman’s Integro-Differential. It is considering the iris to be a perfect circle. The iris boundaries are approximated by two non-concentric circles. And this algorithm boils down to maximizing a specific integro-differential operator that acts as a circular boundary detector. The stronger contrast at the circular region's boundary, the stronger this operator's response[38].

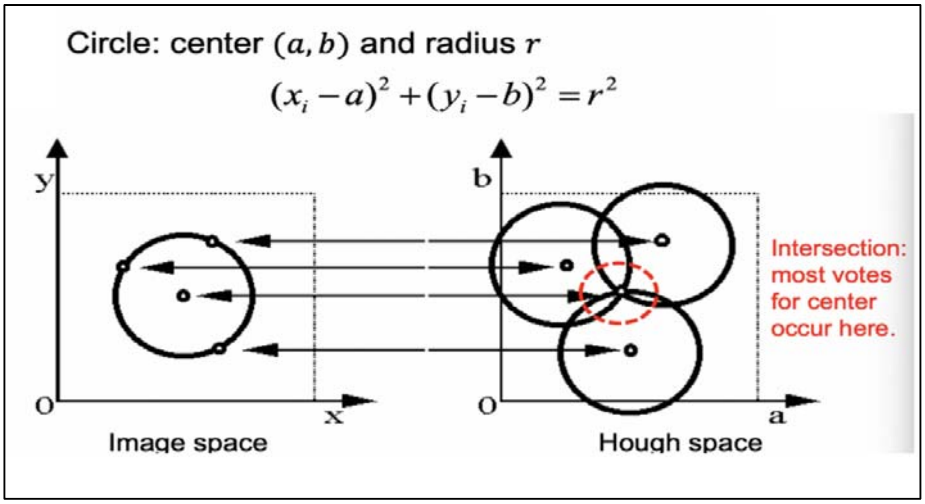
II. Wildes (Hough Transform)

In image analysis, computer vision, and digital image processing, the Hough transform is a feature extraction approach. The technique's goal is to use a voting mechanism to locate imperfect instances of objects inside a given class of forms. This voting mechanism is carried out in parameter space, from which object candidates are obtained as local maxima in an accumulator space that is generated directly using the Hough transform technique. The traditional Hough transform was concerned with detecting lines in an image, but it was later expanded to identifying positions of arbitrary shapes, most often circles or ellipses[39].

The **circle Hough Transform (CHT)** can be applied to detect any circular shape in an image. It's widely used to determine the iris region in an eye image. The equation for finding the circular shape is:

$$(x-a)^2+(y-b)^2=r^2 \qquad \ldots (2.1)$$

where (a,b) is the center of the circle, and r is the radius. If a 2D point (x,y) is fixed, then the parameters can be found according to equation (2.1). The parameter space would be three-dimensional, (a, b, r). And all the parameters that satisfy (x, y) would lie on the surface of an inverted right-angled cone whose apex is at (x, y,θ). In the 3D space, the circle parameters can be identified by the intersection of many conic surfaces that are defined by points on the 2D circle. This operation is done by fixing the radius and then finding the optimal center of circles in a 2D parameter space as shown in Figure (2.4). For each point (x, y) on the original circle, it can define as a circle centered at (x, y) with radius R according to equation (2.1). The intersection point of all such circles in the parameter space would be corresponding to the center point of the original circle[40].


Circle: center (a, b) and radius r
$(x_i - a)^2 + (y_i - b)^2 = r^2$
y
O
x
Image space
b
O
a
Hough space
Intersection: most votes for center occur here.

Figure(2.4): Circular Hough Transform.[41]

Before applying, the Hough transform must use the edge detector. The edge detector is applied to the iris image to generate the edge map which is obtained from the calculation of the first derivative of intensity values and thresholding of the results[42][30].

III. Image Processing Techniques used in the Iris Segmentation stage

Among the image processing techniques that were used in the iris segmentation stage are:

1- Contrast Stretching for Image Enhancement.

Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values. It can only apply a *linear* scaling function to the image pixel values. As a result, the `enhancement' is less harsh. (Most implementations accept a gray level image as input and produce another gray level image as output). Figure (2.5) shows a typical transformation used for contrast stretching. The location of points (r1, s1) and (r2, s2) control the shape of the transformation function.[43]

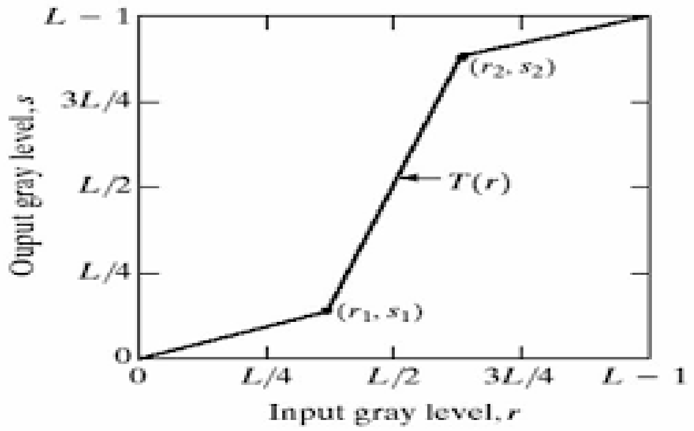


Figure (2.5): A typical transformation for contrast stretching[44]

The contrast of an image is the distribution of its dark and light pixels. An image of low contrast has a small difference between its dark and light pixel values. The histogram of a low contrast image is usually skewed either to the left (mostly light), to the right (mostly dark), or located around the right (mostly gray). Figure (2.6) shows the effect of contrast stretching on the image. It computes the highest and the lowest pixel intensity values, sets them to 255 and 0 respectively, and scales all other pixel intensities according. This technique works best when the histogram of the image is Gaussian[45].

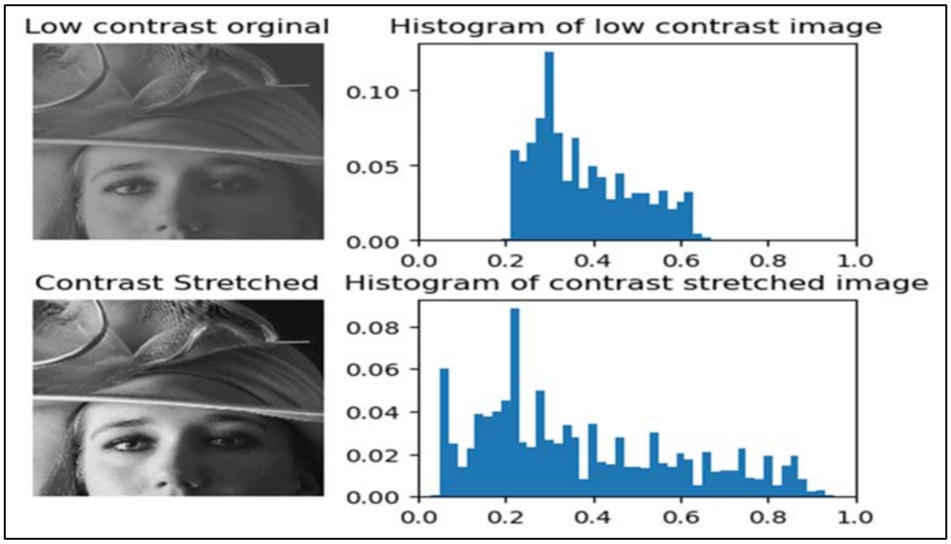


Figure (2.6): the effect of contrast stretching on the image[46]

2- Canny Edge Detector

Many machines vision and image processing algorithms, including those for image enhancement, image segmentation, and etc., start with an essential pre-processing step called edge detection[47][48]. Edge detection is a technique to finding the boundaries of objects within images by detecting discontinuities in brightness. Common edge detection algorithms include Sobel, Roberts, Perwitt, Canny, and Fuzzy logic methods.

The use of Canny edge detector is due to its high performance. This performance is due to providing the ability to adjust the output quality by manipulating edge detection parameters, Sigma and Threshold. This operator is developed by John F. Canny in 1986 [49].

The Canny edge operator algorithm keeps running in five separate steps:

A- **Smoothing**: this step includes applying a Gaussian filter in order to remove the noise.

B- **Finding Gradients**: find the intensity gradients of the image.

C- **Non-maximum suppression**: apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection.

D- **Double thresholding**: the double threshold is applied to determine potential edges

E- **Track edge by hysteresis**: finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

3- Gamma Correction

Gamma Adjustment is a nonlinear operation used to encode and decode luminance or tristimulus values in images. It is represented in the simple cases, defined by the following power-law expression:

$$\boldsymbol{G_{Img(i,j)}} = \boldsymbol{c} \times \boldsymbol{E_{Img(i,j)}}^{\gamma} \qquad \dots (2.2)$$

Where the non-negative real input value $\boldsymbol{E_{Img(i,j)}}$ is raised to the power γ and multiplied by the constant **c** to get the output value $\boldsymbol{G_{Img(i,j)}}$. In the common case of **c**=1, inputs and outputs are typically in the range [0 – 1]. Varying Gamma (γ) obtains family of possible transformation curves as shown in Figure (2.7) [53].

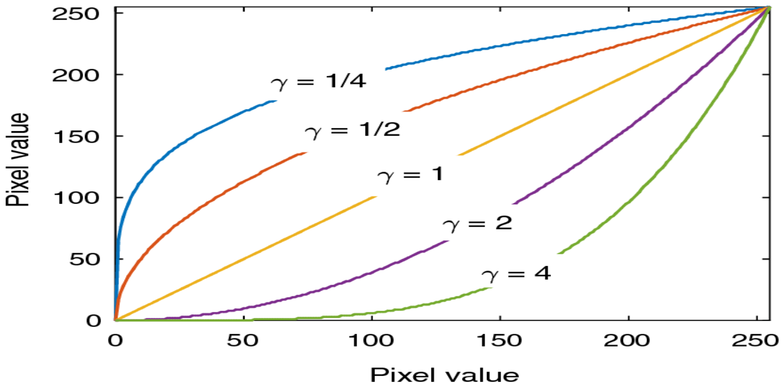


Figure (2.7): Gamma correction curve for different values of γ.

4- Gaussian Filter

Gaussian filter plays an important role in filtering different kinds of surfaces. The simplicity of the algorithm, ease of implementation, and the robustness of the results make this type of filtration the first choice for filtration in any application. The liner Gaussian is very popular in surface characterization, it has been widely used in image processing applications, and it has become an industrial filtration standard. A Gaussian filter can be applied to the input surface by convolving the measured surface with a Gaussian weighting function. In the Gaussian filter, the process is to apply the 2D kernel to the original image $G - Img\ at\ location\ x, y$. The kernel coefficients of Gaussian decrease, when the distance from the center of the kernel increase. Moreover, the center values of the kernel hold more weight than the values on the edges of the kernel.

$$G - Img(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \dots (2.3)$$

Equation (2.3) shows the Gaussian filter. Where σ is the standard deviation of the distribution. The larger the value of σ, the greater the blurring is [50].

5- Morphology Operations

Mathematical morphology is a useful technique in the description and representation of region forms, such as boundaries and convex objects, and for extracting image components[51].The basic morphological operations include dilation, erosion, opening, and closing. In binary morphology, both dilation and erosion operation involve the convolution of an image and structuring elements (SEs), including disk, line, and diamond- shaped structures. The pixels in a binary SE and image (I) are (0) or (1) respectively. As shown in equations (2.4) (2.5).

$$Dilation(Image\ I): I \oplus B = \{x | (B_x) \cap A \neq \phi\} \dots \quad (2.4)$$

$$Erosion\ (Image\ I): A \ominus B = \{x | (B_x) \subseteq A\} \dots \quad (2.5)$$

The opening operation, which combines erosion and dilation techniques, cleans out isolated noisy pixels and flattens image shapes. However, the closing operation or the opposite operation order is useful for bridging minor gaps and filling small holes in the connected region [52]. Equation (2.6) (2.7) represents opening and closing operation.

$$Opening(Image\ I): I \circ B = (A \ominus B) \oplus C \quad \dots (2.6)$$

$$Closing(Image\ I): I \bullet B = (A \oplus B) \ominus C \quad \dots (2.7)$$

6- Binary Object Features

For pattern recognition, image analysis, and computer vision, the quantity of objects, their contours, and their fundamental form characteristics, such as their areas, perimeters, circularities, and centroids, are crucial.[53]. We present several useful features for binary objects. A binary object, in this case, is a connected region within a binary image f(x,y),which will be denoted as $O_{i,}$ I > 0. Mathematically, we can define a function $O_i(x,y)$ as follows:

$$O_i(x,y) = \begin{cases} 1 & if\ f(x,y) \in O_i \\ 0 & otherwise \end{cases} \quad \ldots \ (2.8)$$

1- Area

The area of the ***i***th object $O_{i,}$ measured in pixels, is given by the equation[54]:

$$A_{i=} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} O_i(x,y) \quad \ldots \ (2.9)$$

2- Centroid

The coordinate of centroid (center of area) of object $O_{i,}$ denoted $((x_i, y_i)$ are given by the equations:

$$x_i = \frac{1}{A_i} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} xO_i(x,y) \quad \ldots (2.10)$$

And

$$y_i = \frac{1}{A_i} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} yO_i\,(x,y) \quad \ldots \ (2.11)$$

Where A_i is the area of the object $O_{i,}$ as defined in equation (2.9).[55]

2.4.3 Iris Normalization

The area enclosed by the inner and outer boundaries of an iris can vary due to the dilation and contraction of the pupil. The effect of such variations needs to be minimized before comparing different iris images. To this end, the segmented iris region is typically mapped to a region of fixed dimension. Daugman proposed the usage of a rubber sheet model to transform the segmented iris into a fixed rectangular region. This process is carried out by re-mapping the iris region, I(x,y), from the Cartesian coordinate (x,y) to the dimensionless polar coordinates (r,θ), and can be mathematically expressed as,

$$I\big(x(r,\theta),y(r,\theta)\big) \rightarrow I(r,\theta)\ldots \quad (2.12)$$

Where r is in the unit interval [0,1], and θ is an angle in the range of [0,2]. X(r,θ) and y(r,θ) are defined as the linear combination of both pupillary (x_p(θ),y_p(θ)) and limbic boundary points (x_s(θ),y_s(θ)) as,[17]

$$x(r,\theta) = (1-r)x_p(\theta) + rx_s(\theta) \qquad \ldots (2.13)$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_s(\theta) \qquad \ldots (2.14)$$

Figure (2.8) explains Dougman's Rubber Sheet model, the circular shape represents the segmented iris region and the rectangular shape represents normalized iris which is equivalent to the segmented iris region.

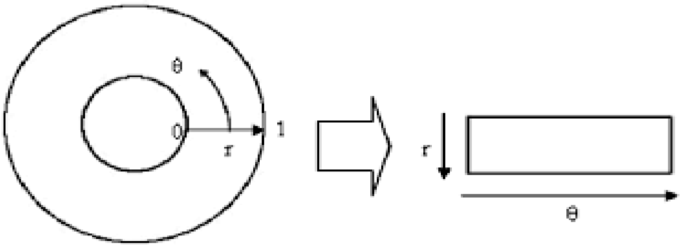


Figure (2.8): Daugman's rubber sheet model.

One of the benefits of normalization is the ease of dealing with the iris area after converting it to a rectangle with a fixed dimension, as we explained previously. In addition to the other benefit of normalization is that the rotations of the eye (e.g., due to the movement of the head) are reduced to simple translations during matching[17].

2.4.4 Features Extraction

Feature extraction is the process of extracting highly discriminative features from the images[6], and in every confirmation method is the crucial assignment. The most important step in achieving high authentication rates in the iris identification system is selecting an efficient feature extractor component. Every iris image has a unique property that sets it apart from the others[24].These features are extracted from the iris image for correct authentication purposes. The iris features can be extracted through the two types of algorithms, hand-crafted algorithms such as 2D Gabor filters, Wavelet transform, Laplacian of Gaussian filters…etc., and Deep learning algorithms such as Convolutional Neural Network (CNN)[6].

The Deep Learning algorithm can be based on the pre-trained models CNN to extract the features such as Inception V3, VGG-Net, Dence-Net, Alex-Net, and etc. It is possible to build a model and train it on the database used by the researcher and extract the features as described in Figure (2.9).

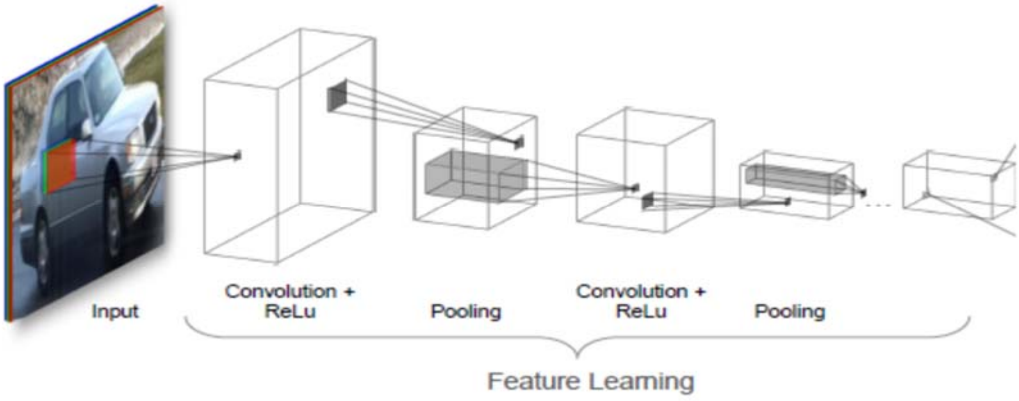


Figure (2.9): Using Deep Learning as Features Extractor.

Before going into the details of the machine learning, artificial neural networks (ANN), Deep learning techniques, and Convolutional Neural Network (CNN), let's get an idea of Fusion process of features and its types

I.Features Fusion

Before delving into the fusion of features, let’s get to know a more general concept, which is biometric fusion. Biometric fusion is the technique of combining the information from each single biometric source together to improve the performance of a biometric recognition system. In the biometric field, the information of the biometric traits can be merged at different levels of fusion as follows:

(a) **pre-classification fusion techniques**: in these techniques, the biometric information is fused before the classification task. This category has two types of techniques which are sensor-level and feature-level fusion techniques as shown in figure (2.10)[21].

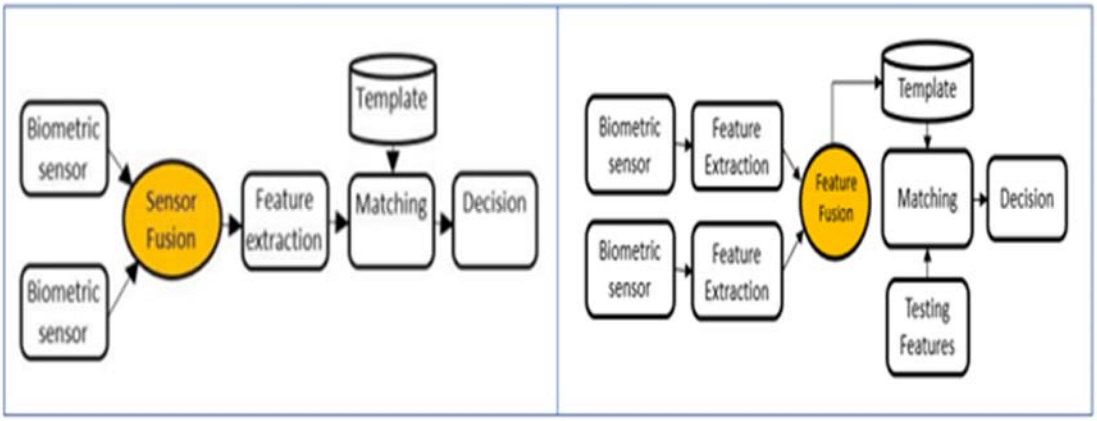


Figure (2.10): Types of pre-classification fusion techniques [56]

-Sensor-level Fusion Techniques

It involves combining sensor raw data. It can be done by collecting the same biometric using two compatible sensors or by using the same sensor for two acquisitions [21].

-Feature-level Fusion Techniques

Features can be extracted from the same (or different) input image and combined into a feature vector which is stored in a template. The new feature vector represents the identity of the individual in a different (and more distinct) feature space. Compared with other fusion techniques, feature-level fusion is referred to as tightly convergent integration, and is expected to achieve better accuracy in personal identification[57] .

(b)post-classification fusion techniques: the biometric information in these techniques is fused after classification task based on the scores obtained by the classifiers as shown in figure (2.11) [21].

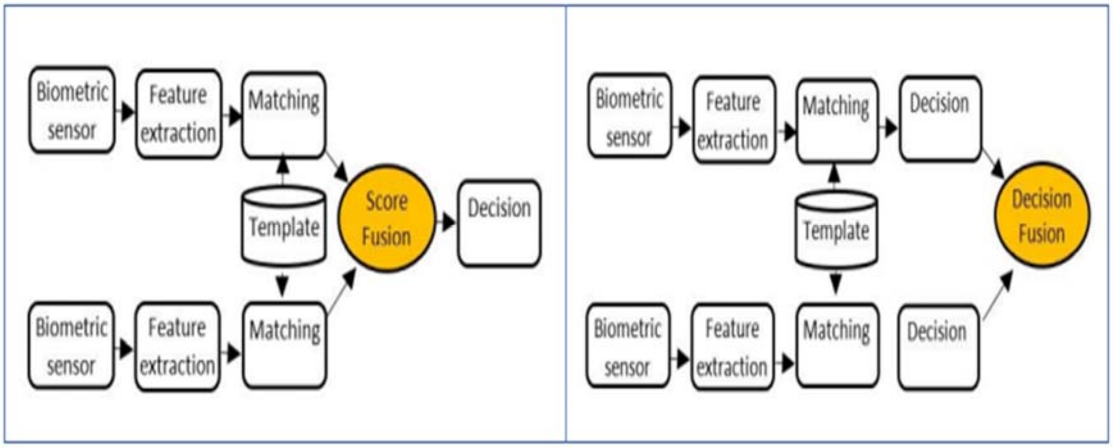


Figure (2.11): Types of post-classification fusion techniques [56]

-*Matching score level-Fusion Techniques*

The matcher of each biometric system, implies a similarity score, which indicates the proximity of the input feature vector with the stored template feature vector. These scores can be combined to get the final decision.

-*Decision level-Fusion Techniques*

Each biometric system makes its own identification decision. A majority vote scheme can be used to make the final identification decision[57].

II. Machine Learning Models

A branch of artificial intelligence (AI) that enables computer programs to learn from data and then make appropriate decisions based on the information that has been learned from a prior experience. It depends on computer science, statistics, and mathematic. Machine learning methods are generally classified into three classifications [58]:

- **Supervised learning**: adjusts a system such that it generates the desired output given the supplied data. The learning data consists of two tuples, attributes and label, where attributes refer to the input data and label to the desired result. The objective is to modify the system so that it can forecast the desired output in response to a new input.

- **Unsupervised learning**: includes data that consists of input vectors without any target output. In unsupervised learning, there are different objectives such as clustering, visualization, and density estimation.
- **Semi-supervised learning (reinforcement learning):** uses unlabeled data to first develop a feature representation of the input data, which is subsequently used to the supervised job. The data samples with corresponding labels and the data samples for which the labels are unknown can be separated into two categories in the training dataset. Sometimes referred to as reinforcement learning, semi-supervised learning can involve not giving each time an explicit form of error but just receiving a generalized reinforcement that indicates how the system should change its behavior.

III. Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are computer techniques that imitate how the human brain does a particular activity through intensive processing of fundamental units that are closely coupled in groups. These units are the computational components that have the characteristics of neurons or nodes. By adjusting weights, experimental data and practical knowledge are processed and made available to the user[59]. There are many different kinds of ANN, including Multi-Layer Perception (MLP), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs), that have gained the most popularity in recent years. Among the concepts and techniques of ANNs that it will be explained in this section are:

1- ***Multi-Layer Perceptron (MLP) Structure*** is an ANNs feedforward structure. A number of computing units (nodes) with trainable weights and biases at various levels make up the major parts of an MLP (one input layer, one or more hidden layers, and one output layer). As a result, as shown in Figure (2.12), every node (neuron) in these layers is connected to every node in the subsequent layer. The MPL, which is trained using backpropagation and forward-propagation, will use a feedforward method to translate an input vector (X) to the desired output vector (Y). By using the former, the biases and weights of the network are determined to roughly approximate the complex relationship between the desired output and the input vector, and the training's goal is to reduce the discrepancy between the predicted results and the desired results by using a specific cost function. This process is repeated until the difference reaches zero or almost zero[60].

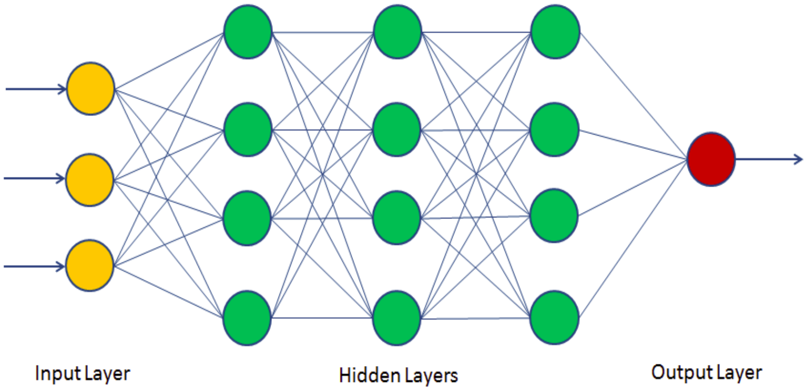


Figure (2.12): the basic architecture of the Multi-Layer Perceptron.

A neuron first computes the dot multiplication between the input vector $X = [x_1 x_2 \dots x_n]$ and its corresponding weights $W = [w_1 w_2 \dots w_n]$. and then adds the bias value(b) as shown in the equation (2.15). The value (Z) is finally passed on to the activation function[61].

$$Z = \sum_{i=0}^{n} x_i w_i + b \qquad \dots (2.15)$$

2- ***The activation Functions:*** A neural network must have the activation function in order to learn and complete any challenging tasks. There are many different types of activation functions, some of which are linear and some of which are nonlinear. The output value for a unipolar activation function is either (0 and 1) or (-1 and 1) for a bipolar activation function[62]. The most widely used activation functions in hidden layers of neural networks are the sigmoid function and the Rectified Linear Unit (ReLU) function. The equations (2.16) and (2.17) explain the sigmoid function and ReLU function respectively.

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \dots (2.16)$$

$$R(z) = \max(z, 0) \qquad \dots (2.17)$$

In deep networks, the most successful activation function is ReLU. It is a non-linear function that replaces all image pixels with zero values in the activation map when their values are negative. As a result, calculation speed is an advantage and overfitting are less common[58].

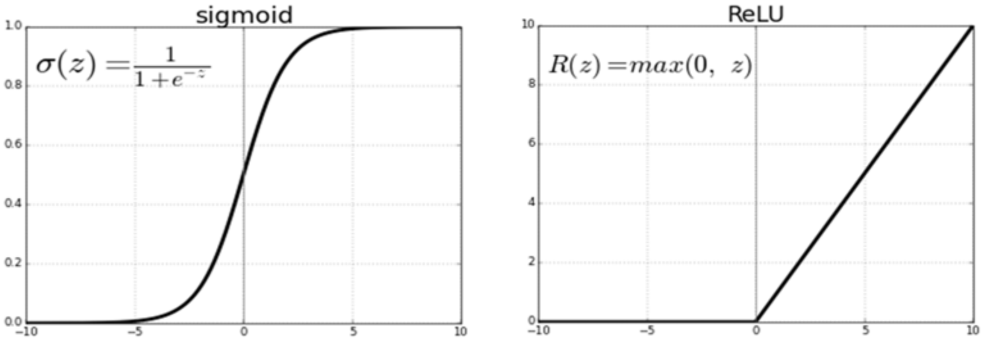


Figure (2.13): Activation Function (Sigmoid and ReLU).

A ***SoftMax activation function*** is a mathematical function that takes an input vector and produces an output vector with items that add up to 1 in the range (0–1). In other words, the SoftMax function's total outputs add up to 1. SoftMax function is important in the final layer of deep neural networks for multiclass classification because its outputs are probability distributed. SoftMax function is implemented by applying equation (2.18), where the variable *y* is represented the number of classes[58]:

$$Softmax(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \qquad \text{... (2.18)}$$

3- ***Loss Functions*:** When a deep network is being trained, the error (difference between the prediction and the ground truth label) is calculated using a loss function. There are numerous loss functions that can be employed in deep learning networks, depending on the application. For instance, in classification problems, mean squared error loss, cross-entropy loss, and hinge loss are frequently employed. Regression problems are well-suited to absolute deviation error loss [58]. Most common functions:

-Mean Squared Error (MSE) Loss

Mean Squared Error (MSE) loss function, the most used loss function in machine learning, computes the squared average error E of all the individual errors. Mean Squared Error (MSE) can be calculated by:

$$E = \frac{1}{n}\sum_{i=1}^{n} {e_i}^2 \qquad \text{... (2.19)}$$

where e_i is the ***i***th output neuron's individual error, denoted by:

$$e_i = target(i) - output(i) \qquad \text{... (2.20)}$$

During the training phase, an error is calculated using a loss function at the output layer, and its derivative (gradient) is propagated in the network's reverse direction. The network's weights are then updated with the appropriate gradients [58].

-Cross-Entropy Loss

Another loss function that is frequently utilized in regression and classification issues is cross-entropy loss. The cross-entropy loss function is chosen for SoftMax classifiers since it is employed when the output is a probability distribution. This function applied by:

$$H(y) = -\sum_i y_i' \log(y_i) \qquad \dots (2.21)$$

Where y_i represents the output of the classifier, y_i' represents the target label[58].

4- ***Back-propagation Algorithm:*** Due to its effective and simple structure, the supervised learning algorithm known as back propagation is frequently employed for training neural networks (NNs). To reduce a neural networks prediction error, the gradient descent technique is being used[59].After setting all weights and biases in the network to a random beginning value *to begin the training, there are two phases:*

- ***Forward Propagation***: Equations (2.15) and (2.17) are used to propagate input instances across the entire network layer by layer from input to output in order to create a prediction value.
- ***Backward Propagation***: The gradient descent learning algorithm, which works as follows, starts at the output and goes back to the input. When utilizing loss equation (2.21) to calculate the error of the output layer, the delta $\delta_J{}^L$ is obtained, where J is the index of the neuron in that layer and L is the layer as following:

$$\delta^L = \nabla_x \mathcal{L} \odot g'(Z^L) \qquad \dots (2.22)$$

Where:

δ^L: Matrix of Deltas of all neuron in L.

$\nabla_x \mathcal{L}$: Gradient of Loss with respect to x.

X: Output of Activation Function.

$\odot$: Hadamard product (element-wise product of matrices).

In lower hidden layers the deltas are computed using the following equation:

$$\delta^{l} = ((w^{l+1})^{T}\delta^{l+1}) \odot g'(z^{l}) \qquad \text{... (2.23)}$$

Where:

l: Number of Hidden Layers.

$(\boldsymbol{w^{l+1}})^{\boldsymbol{T}}$***:*** Weight Matrix of the next layer.

During training, neural networks (NNs) learn by adjusting weights as follows:

$$\Delta \mathrm{w}_{\mathrm{i,j}} = -\eta \frac{\delta \mathcal{L}}{\delta \mathrm{w}_{\mathrm{i,j}}} = -\eta \delta_{\mathrm{j}} \mathrm{z}_{\mathrm{i}} \qquad \text{... (2.24)}$$

Where:

η: learning Rate.

δ_j: Delta of neuron j.

When these equations are repeated for all input instances, this is called an epoch. Neural networks (NNs) continue in many epochs until we get the optimal solution[57]. Learning rate is hyper-parameter which cannot be too high or too low. Large value of learning rate can miss the optimum value, and too low learning rate will result in slow training time [58].

5- ***Optimization Algorithms:*** is the process of developing a procedure that can minimize or maximize the output of function, giving it the same input. By modifying some variables such as weights and biases of the network. This is a difficult task since there are large and high dimensions space to seek the optimal values for these variables.

Gradient descent is an optimization technique used to minimize or maximize the cost function by calculating gradient necessary to update the parameters. There are many performance optimization tools used in deep neural networks, such as

Batch Gradient Descent (BGD), Mini-Batch Gradient Descent (M-BGD), Stochastic Gradient Descent (SGD).

1-Batch gradient descent (GD): The weight parameter is updated after computing the error gradient with respect to the weight parameter *w* for the full training set in classical gradient descent (GD), also known as batch gradient descent as shown:

$$w = w - \eta \, . \nabla \mathrm{E}(w) \qquad \text{... (2.25)}$$

Where:

w: weight.

η: learning rate.

$\nabla \mathrm{E}(w)$: error gradient with respect to weight *w.*

Batch gradient descent can be extremely slow and is infeasible for datasets that do not fit in memory since we must calculate the gradients for the entire dataset in order to make just one update. Additionally, batch gradient descent prevents us from updating our model online [63].

2-Stochastic gradient descent**:** In contrast, stochastic gradient descent (SGD) updates a parameter for each training example $x^{(i)}$ and label $y^{(i)}$:

$$w = w - \eta \, . \nabla \mathrm{E}(w; x^{(i)} y^{(i)}) \qquad \text{... (2.26)}$$

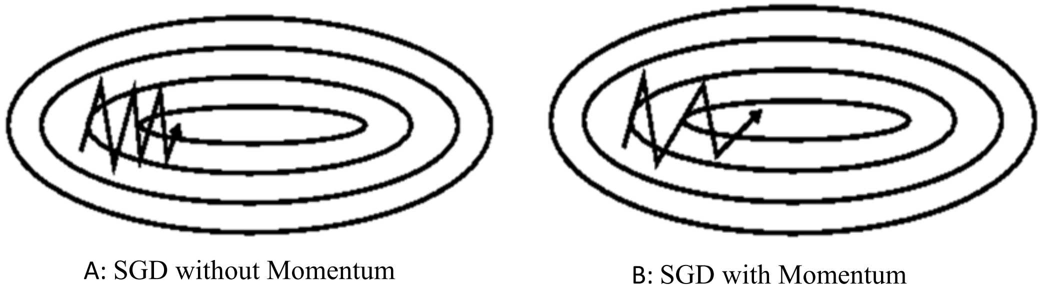
Where:

$\nabla \mathrm{E}(w; x^{(i)} y^{(i)})$: gradient of loss function.

For big datasets, batch gradient descent conducts redundant calculations since it recalculates gradients for related samples before to each parameter update. SGD eliminates this redundancy by carrying out updates one at a time. As a result, it is typically significantly quicker and can be used for online learning.

Momentum

SGD has difficulty negotiating ravines, which are frequently found around local optima and are defined as regions where the surface curves substantially more sharply in one dimension than another. SGD oscillates across the ravine's slopes while moving slowly towards the regional optimum towards the bottom, as seen in Figure (2.14)



A: SGD without Momentum

B: SGD with Momentum

Figure (2.14):The effect of momentum [63]**.**

In essence, momentum is used to push a ball down a hill. As it rolls downhill, the ball gains momentum and accelerates (until it achieves its terminal velocity, which is 1 if there is air resistance). The momentum term affects our parameter updates in a similar manner, increasing them for dimensions whose gradients point in the same directions and decreasing them for dimensions whose gradients vary. We benefit from quicker convergence and less oscillation as a result.

3-Mini-batch gradient descent

Finally, mini-batch gradient descent combines the best aspects of both methods by updating each mini-batch of ***n*** training examples.

$$w = w - \eta \, . \nabla E(w; x^{(i:i+n)} y^{i:i+n}) \qquad \text{... (2.27)}$$

This approach lowers the variance of parameter updates, which may result in more steady convergence. It can also make use of highly optimized matrix optimizations seen in modern deep learning libraries, which greatly increase the efficiency of computing the gradient in a mini-batch. When training a neural network, mini-batch gradient descent is frequently utilized, and when mini-batches are used, the term SGD is also frequently used[63].

IV. Deep Learning

Deep learning is a new field of machine learning which has gained a lot of importance lately. In order to learn various features with various levels of abstraction, deep learning refers to architectures that incorporate several hidden layers (deep networks). With higher level learnt features expressed in terms of lower-level characteristics, deep learning algorithms strive to take advantage of the unknown structure in the input distribution to find good representations, frequently at several levels. For many years, developing a pattern recognition or machine learning system required a great deal of domain knowledge and careful hand engineering to create a feature extractor that converted the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning system, such as a classifier, could detect or classify patterns in the input. Deep learning algorithms are capable of learning the proper set of features, and they do this far more effectively than manual coding would. Deep learning entails automatically picking up these features while being trained, as opposed to manually creating a set of rules and algorithms to extract them from raw data[58].Figure (2.15) explains the deep network architecture.

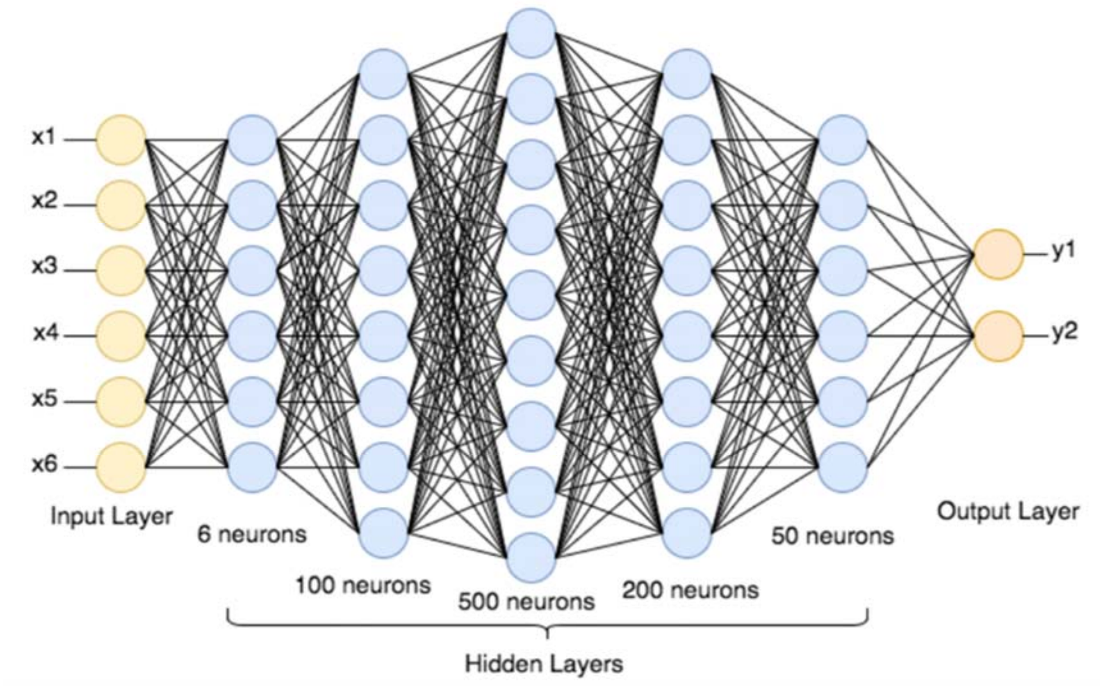


Figure (2.15): Deep Network Architecture[64]

Deep learning can be classified into:

1- **Deep Neural Network (DNN).**
2- **Convolutional Neural Network (CNN).**

1- Deep Neural Networks (DNN)

The DNN is a multi-layered hidden neural network created artificially. The ANN architecture that is utilized for DNN the most frequently is (MLP) Neurons that are coupled at different levels together to form neural networks. The implementation of DNN requires lengthy computation times and numerous data feeds for training samples, therefore the number of weights in these networks will be in the thousands or millions [59].

2- Convolutional Neural Network (CNN).

An advanced form of artificial neural network known as a convolutional neural network uses the mathematical operation known as convolution for generic matrix multiplication in at least one of its layers. They are employed in image detection and processing since they were created expressly to handle pixel data[39].

An input layer, hidden layers, and an output layer make up a convolutional neural network. Any middle layers in a feed-forward neural network are referred to as hidden layers since the activation function and final convolution hide their inputs and outputs. The hidden layers in a convolutional neural network contain convolutional layers. This typically contains a layer that does a dot product of the input matrix of the layer with the convolution kernel. The convolution procedure develops a feature map as the convolution kernel moves across the input matrix for the layer, adding to the input of the following layer. Following this, further layers like normalizing, pooling, and fully connected layers[65] .Figure (2.16) shows CNN layers.

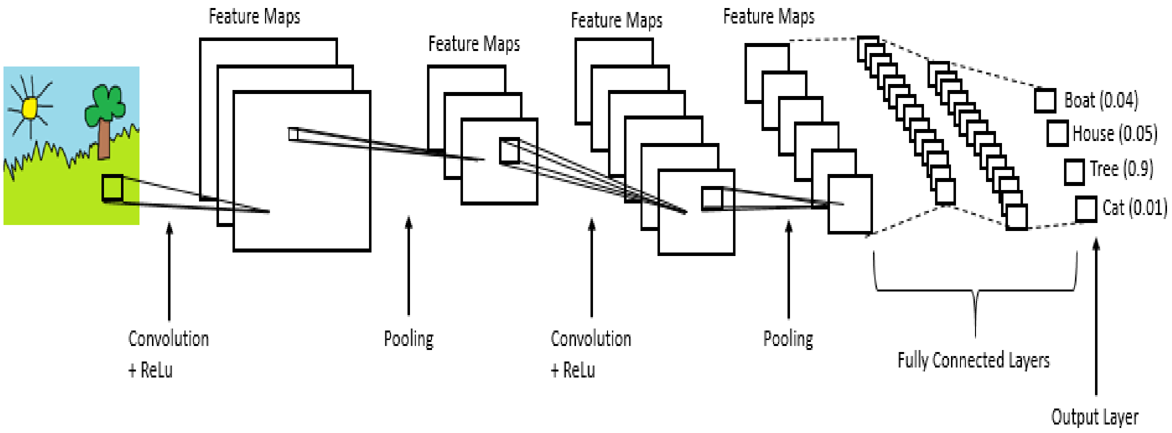


Figure (2.16): CNN Architecture[66]

In CNN, there are different types of layers. Each layer works in a specific way, for a specific task. These layers are:

- **Convolutional Layer.**

A convolutional layer is the main building block of a Convolutional Neural Network (CNN). It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image and creates an activation map or feature map as shown in the Figure (2.17) [67]

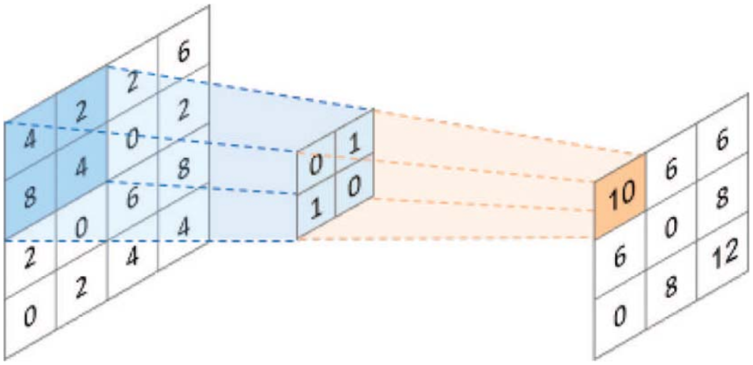


Figure (2.17): The Convolutional operation between the input layer and the Kernel in CNN.

For one input, several convolution filters are used. The final output for a single convolutional layer is then created by combining the resulting activation maps, and this final result serves as the following layer's input data. Each value of the filter matrix represents a default weight. To impart distinct qualities or features to each matrix from the feature map matrices, these values must differ from one filter to the next[68]. A general convolutional layer's process for images described in the following equation [69]:

$$Z_{ij} = (X * K)_{(i,j)} = \sum_{l=1}^{K1} \sum_{a=1}^{K2} \sum_{b=1}^{C} X_{(i+l,j+a,b)} K_{(l,a,b)} \dots (2.28)$$

Where:

X: Input.

K: Kernel Weights.

Z: Feature Map.

K1: Height of the Kernel.

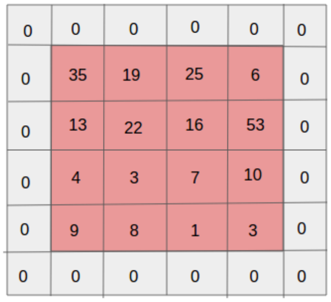
K2: Width of the Kernel.

C: Number of Channels.

The results of each convolutional layer will be passed to the activation function (ReLU) as in the equation (2.18). The size of the feature map is determined by three parameters these are:

1. **Depth:** The depth denotes the number of filters used during the convolution process. The activation maps' "depth" will equal three if the original image was convolved using three filters.
2. **Stride**: indicates the number of pixels in the filter matrix that are moved during convolution. The kernel will advance by 1 if a stride is equal to 1. In the same case, the kernel advances by 2 if a stride is equal to 2. The function maps get smaller as the number of steps rises.

3- **Zero-padding**: The process of symmetrically adding zeroes to the input matrix is known as zero-padding. It's a widely utilized tweak that enables us to change the input's size to meet our needs. When the dimensions of the input volume must be kept in the output volume, it is mostly employed in the design of CNN layers. There are three types of padding in CNNs, these are Same padding, Causal padding, and Valid padding [58].



0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Figure (2.18): A zero-padded 4×4 matrix becomes 6×6 matrix [39].

- **Batch Normalization Layer**

Batch Normalization is a normalization technique done between the layers of a Neural Network instead of in the raw data. It is done along mini-batches instead of the full data set. It serves to speed up training and use higher learning rates, making learning easier.

- **Max Pooling Layer**

It involves down sampling a collection of nearby pixels into a single pixel. To make the image activation maps smaller, this layer was applied after the convolution layer. There are two types of pooling used in CNNs: Max pooling and Average pooling. The max-pooling chooses the largest element from the corrected Activation Maps within each window, determining a spatial area (sub-region) such as (a 2×2 window). This down sampling reduced the Activation Maps picture size from (4×4) to (2×2) as shown in figure (2.19). Whereas average pooling returns the mean value for each sub-region [70].

The max pooling is preferred over average because it is more efficient and stronger and it can be described as an equation:

$$S_i = \max_{i \in R_j} h_i \qquad \dots (2.29)$$

Where h represents pixels in the window Rj from the rectified Activation Maps. The importance of the pooling process is that it doesn't only reduces the dimensions of the activation map, but also solves the problem of overfitting.

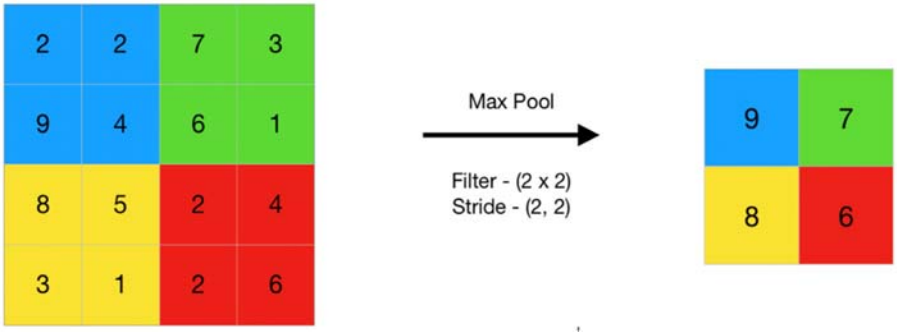


Figure (2.19): Max-Pooling Operation in CNN[71]

- **Flatten Layer**

The generated feature maps are typically flattened, or converted into a one-dimensional (1D) array of vectors (or numbers), and then connected to one or more completely connected layers (normal Artificial Neural Network). The primary goal of the flatten layer is to make the input layer of the CNN's classification component suitable for receiving the signal from earlier feature extraction layers. Figure (2.20) demonstrates how to do the flatten operation[72].

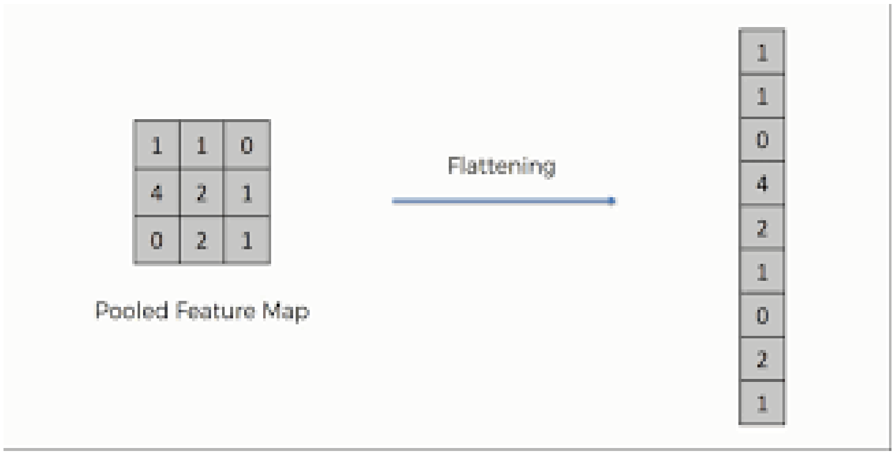


Figure (2.20): Flatten Operation in CNN[73]

- **Fully Connected Layer (Dense Layer)**

Each node in this layer ("Fully Connected") has a full connection to every node in the layer below (similar to the classic MLP network), as shown in Figure (2.21) [89]. Using the SoftMax function in the output layer, it serves as a classifier in the final layers of the CNN structure to determine the liklihood that an object is present in the image[67].

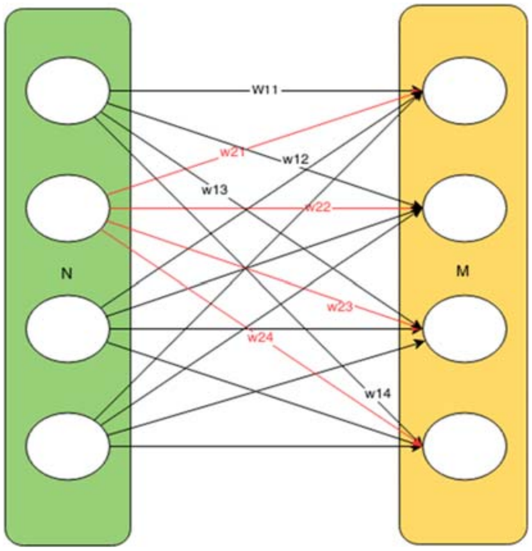


Figure (2.21): Fully Connected Layer in CNN.

V. Regularization Techniques

Many techniques can be used to solve some of the problems that may encounter in the work of CNN, from these problems is, Overfitting Problem. First of all, let’s see the Overfitting Problem.

Overfitting Problem

Deep neural networks may express learning quite effectively. Over-fitting issues in deep neural networks can result from a lack of control over the learning process. Models with overfitting have a limited ability to generalize. Despite the model producing a lot of training or validation data, this causes the test data to be poorly predictable. When there is a significant difference between training and testing errors, it takes place. Figure (2.22) illustrates the overfitting issue. This indicates that the deep neural network model is set up to match the training data well rather than recognize data patterns [74][75].

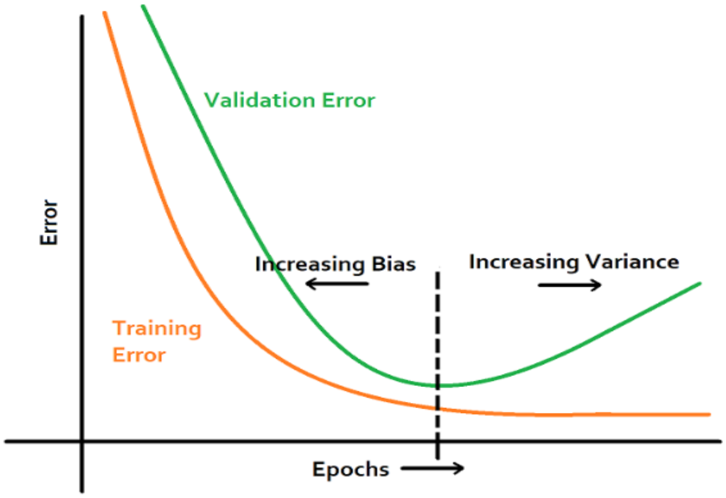


Figure (2.22): Overfitting Problem in CNN.

The techniques to solve this problem:

1. Early Stopping

The ideal number of epochs for deep learning network training is determined by early halting, a potent regularization strategy. The training process must be stopped after a few additional periods using early stopping, as shown in Figure (2.23), because if the number of epochs increases, it will result in overfitting, and if the number of epochs decreases, it will result in underfitting. This is because as the number of epochs increases, the loss increases in the validation sample and reduces in the training sample[76].

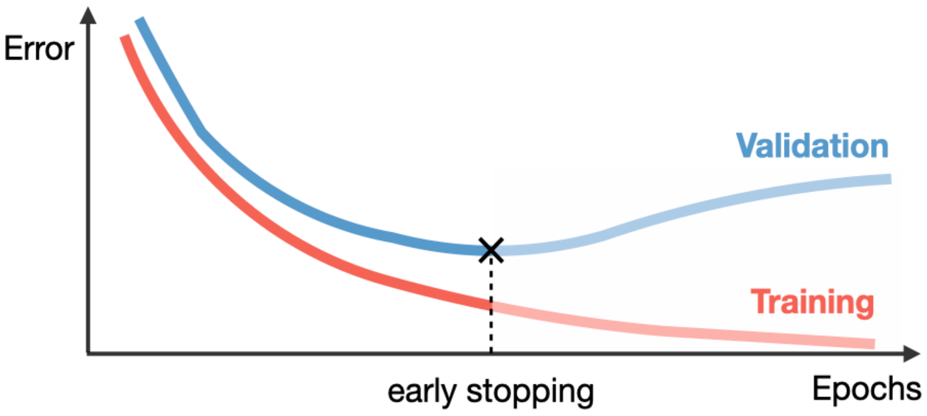


Figure (2.23): Early Stopping Technique.

2. **Data Augmentation**

Data augmentation is a method that increases the number of images used for training the neural network. That is, creating new data for categories that have fewer numbers in the data set. This process succeeds the constraining effect on data to prevent an asymmetric representation and successfully escape overfitting complications. Appropriate data augmentation techniques can help improve deep learning model strength. There are various techniques to augment classic data such as zooming, flipping, shifting, rotation, adding noise, and transformation to be implemented to the original images[77].

3. **Dropout**

The dropout strategy involves randomly removing a group of neurons at each training cycle with a predefined probability value. Nitish Srivastava and Geoffrey Hinton presented this method in 2014, which significantly enhanced the capability of neural networks to address the overfitting issue [78].

2.4.5 Classification

A classifier, also known as a classification model, is used to predict categorical labels and decide which category the data belongs to (decision). There are two approaches for classification: the static formula for matching and classification of Algorithms based on Machine Learning.

I. The static formula for matching

The database templates are compared to the user template using the matching metric. The similarity between two templates was determined by the matching metric. There are several metrics for matching such as Hamming Distance, Normalized Correlation, Manhattan Distance, Minkowski Distance, Weighted Euclidean Distance, and etc. In this thesis, using Weighted Euclidean distance because it is used to measure the distance between two vectors and for the ease of its calculations.

Weighted Euclidean distance

To measure the similarity between two templates, can use weighted Euclidean distance according to equation (2.30), which compares the two templates to identify the class, where each template consists of integer values. The minimum distance means the two templates are matched[6].

$$D(x,y) = \sqrt{\sum_{i=1}^{f} w_i (x_i - y_i)^2} \qquad \text{... (2.30)}$$

Where:

x_i: Value of ith measure for the data.

y_i: Corresponding benchmark value for that measure.

w_i: Value of the Weight $0 < w_i < 1$ and $\sum_{i=1}^{n} 1$.

II. Classification Algorithms based on machine learning

Machine Learning systems are categorized based on how much and what kind of supervision they receive during training. There are a variety of classification methods available, such as Support Vector Machines (SVMs).

Support Vector Machine is one of the supervised machine learning techniques used for regression and classification tasks. But due to its great accuracy across a range of applications, it is mostly employed for classification tasks. The basis of Support Vector Machine (SVM) is the idea of locating an ideal hyperplane (also known as a decision surface) that best separates the input data set into two classes. The margin is the distance between the hyperplane and the nearest point of either of the input data sets (this distance should be equal). Support vectors are the data that are most closely related to the hyperplane, as seen in Figure (2.24).To increase the likelihood that any new data will be correctly classified, the hyperplane with the biggest margin between it and any point in the training data set should be selected.[27]

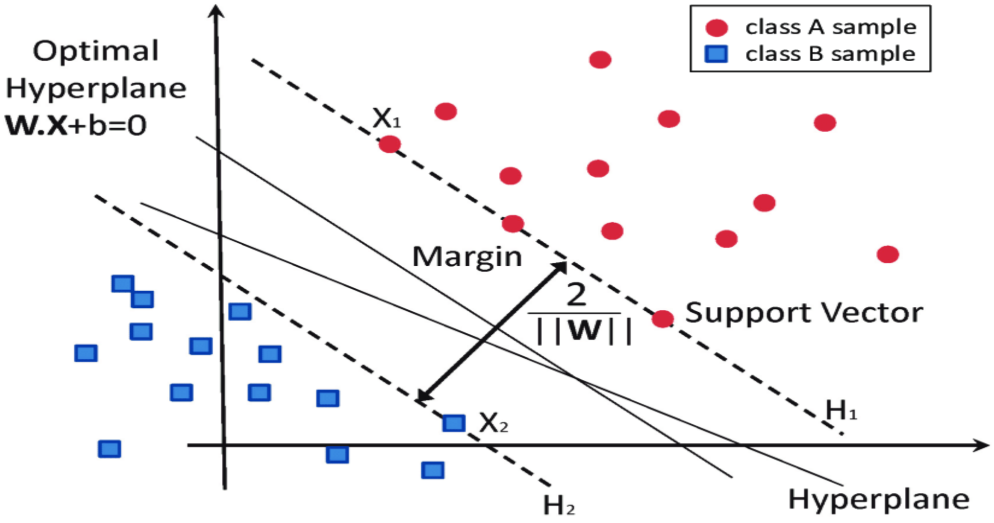


Figure (2.24): Support Vector Machines (SVM).

One benefit of SVM is that it makes decisions with a high degree of efficiency by only using a portion of the training data. This means that the best high level is selected based on the data points that are placed on the supporting vectors rather than on the whole training data. When the number of dimensions is bigger, it also operates. SVM is mostly favored for categorizing images due to these and other benefits[79].

2.5 Performance Measures

In this part, the performance measures that are used to measure the performance and efficiency of classification algorithms will explain [80].

1- ***Accuracy***: the number of accurately identified cases, whether positive or negative, serves as a measure of accuracy.

$$\boldsymbol{Accuracy = \frac{TP+TN}{TP+TN+FP+FN}} \qquad \text{... (2.31)}$$

2– ***Precision***: testing the true positive from the expected positives yields information on the accuracy of the model's performance.

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \qquad \text{... (2.32)}$$

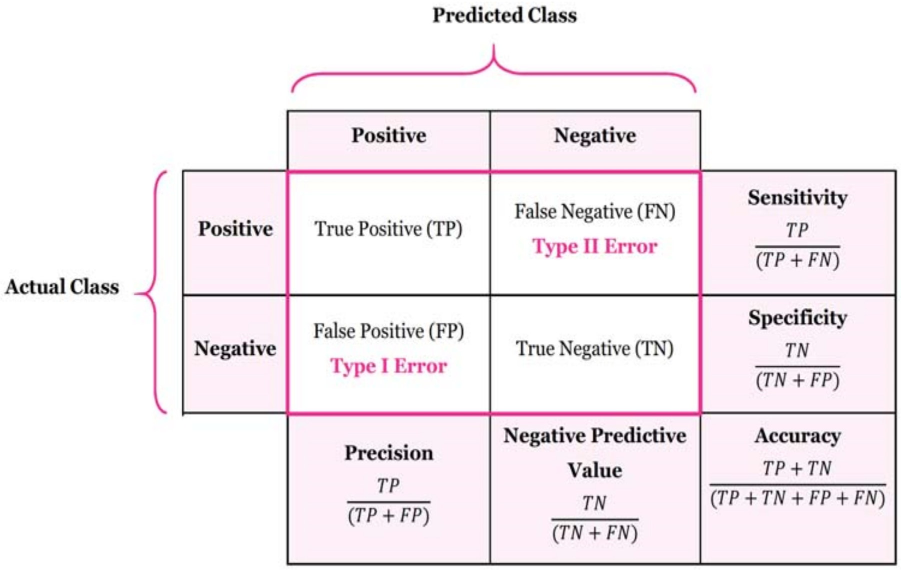
3– ***Recall***: Recall is the accuracy with which positive samples are correctly identified.

$$\boldsymbol{Recall = \frac{TP}{TP+FN}} \qquad \text{...(2.33)}$$

4– ***F-score***: for calculating a balanced mean output, the F1-score demonstrates the combination of precision and recall.

$$\boldsymbol{F - score = \frac{2*Recall*Precision}{Recall+Precision}} \qquad \text{... (2.34)}$$

Figure (2.25) depicts the above parameters.



		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure (2.25): Confusion Matrix

The prediction error is recorded by four parameters [80]:

•**True Positive (TP)** is the positive states that are correctly labeled as positive states.

•**False Positive (FP)** denotes the negative states that are incorrectly labeled as positive states.

•**True Negative (TN)** represents the right classification of negative diagnosis.

•**False Negative (FN)** indicates the positive cases that are incorrectly classified as negative.

Chapter Three

Proposed

Iris Identification System

Chapter Three
Proposed
Iris Identification System

3.1 Introduction

This chapter explains the processes of stages of the Iris Identification System. This system consists of a number of stages: Pre-processing (Iris Segmentation, and Iris Normalization), Features Extraction (using CNN Algorithms), and the classification.

Image enhancement and processing to extract the region of interest is carried out in the pre-processing stage. As for the process of extracting the features, a new model for CNN is built to extract the iris features and pass it to the classification stage. In the classification stage of the proposed system three approaches are used to classify and match data.

The first approach is using artificial neural network (ANN) with SoftMax classifier. The second approach: build a template for each person from the features extracted from CNN and store it in the database. And recognize the person by matching the features using distance. The third approach: input features extracted from CNN on Support Vector Machine (SVM) Classifier.

3.2 Proposed System

In order to develop a model that is capable of classifying images based on the extracted iris features, Figure (3.1) explain the major system stages (a: pre-processing, b: Feature Extraction and Classification by CNN+ SoftMax and, other classification methods that used in this system (Feature Fusion and CNN + SVM)) are followed:

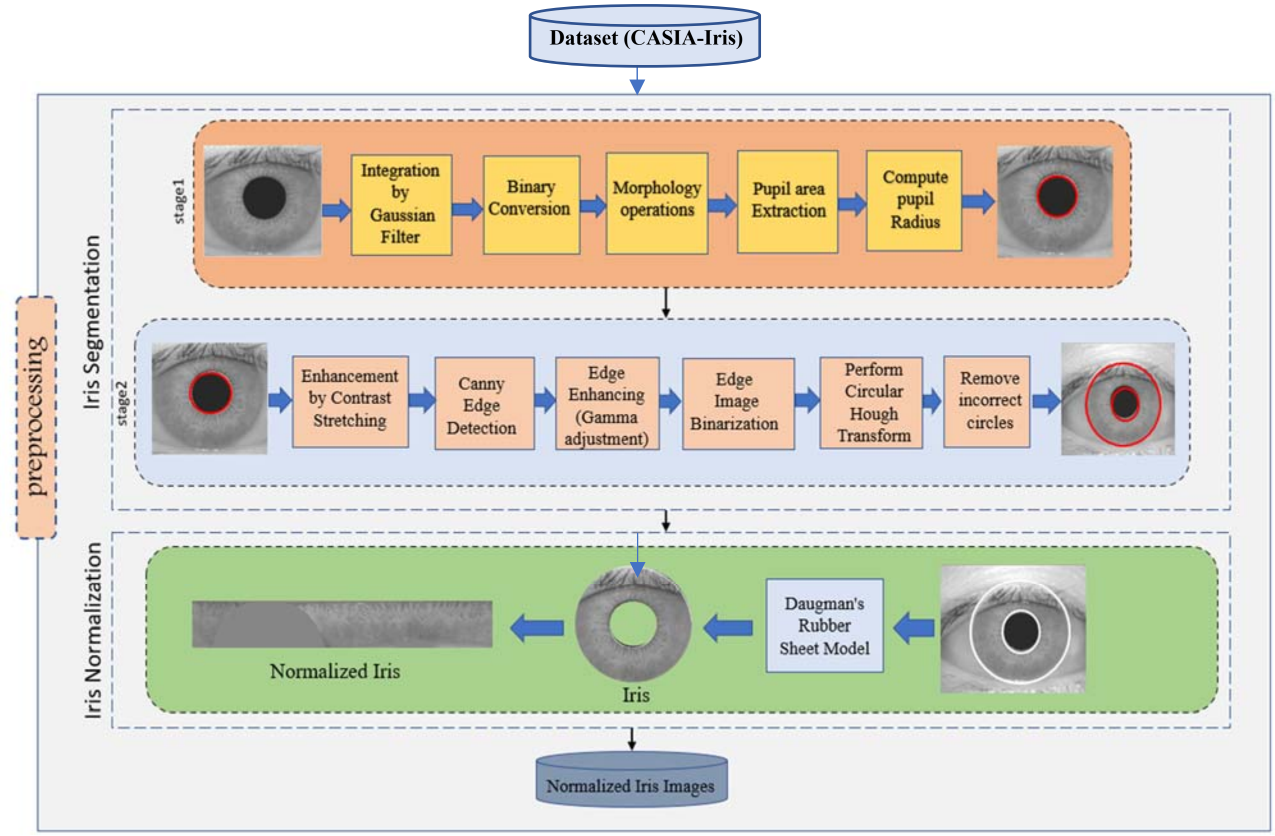


Figure (3.1.a): Pre-Processing Stage for Proposed Iris Identification System

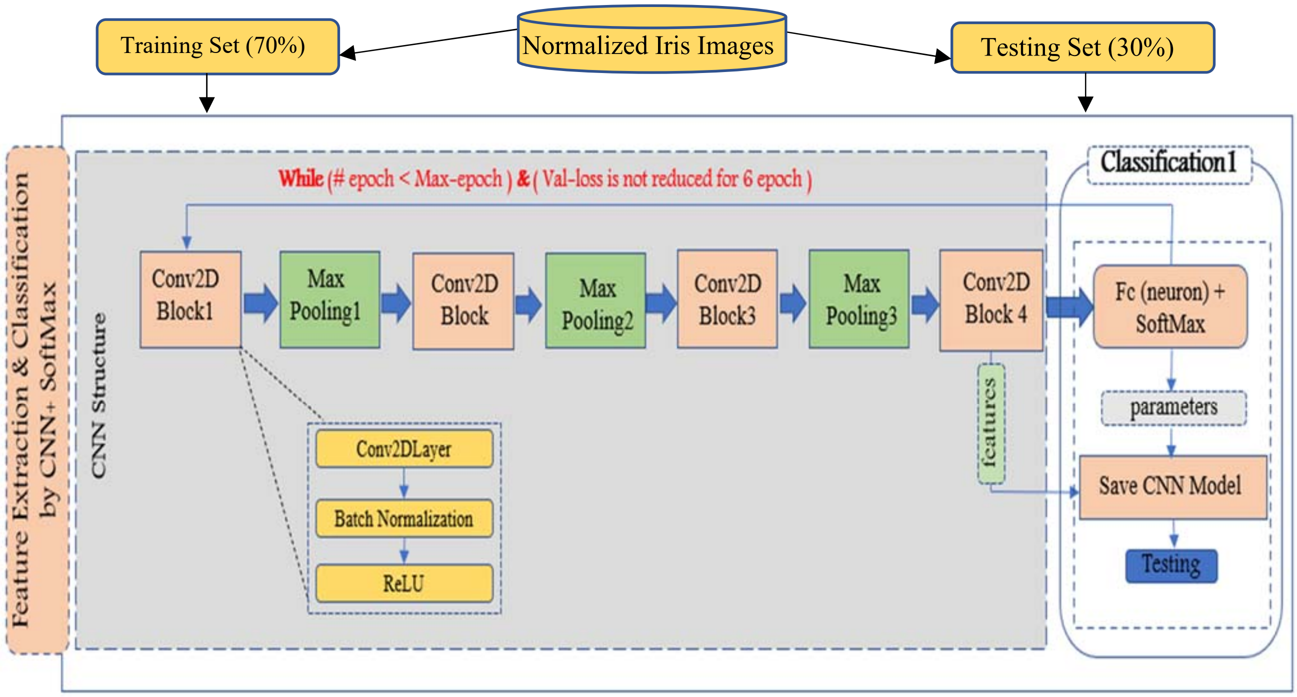


Figure (3.1.b): Feature Extraction Stage for Proposed Iris Identification System

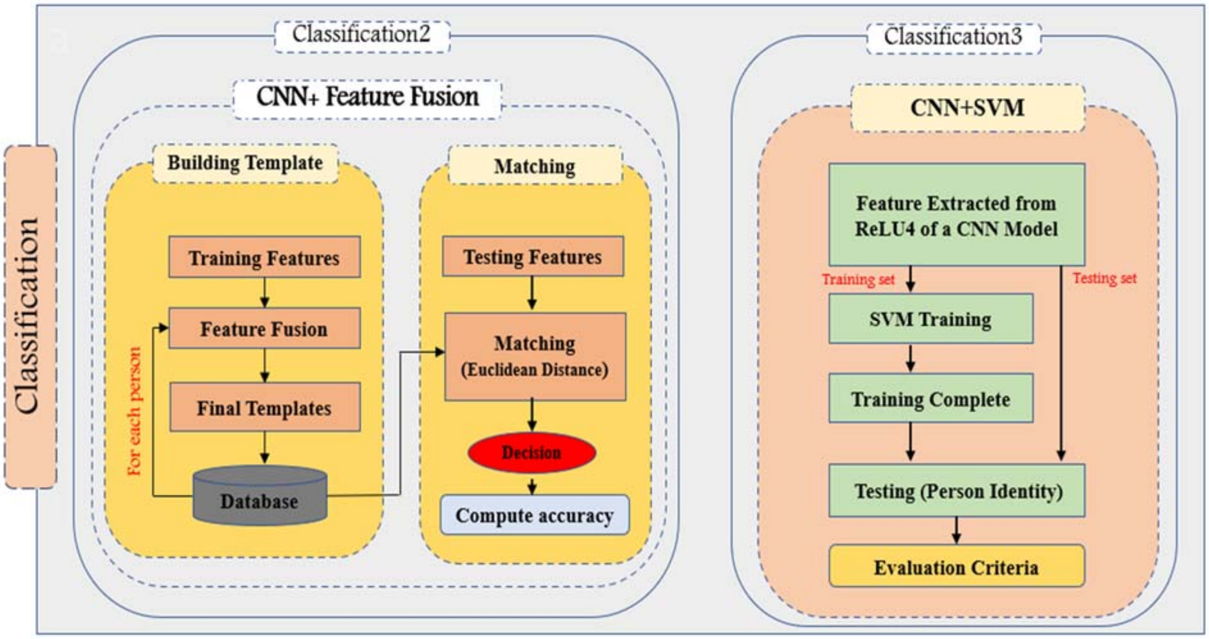


Figure (3.1.c): Classification Methods of the proposed Iris Identification system

3.3 Pre-processing

In Iris Recognition Systems, the process of detect the iris of the eye is an important step as it aims to make an accurate determination of the iris boundaries. As shown in figure (3.2), the iris is an annular part located between the white sclera and the pupil, representing the outer and inner circles respectively.

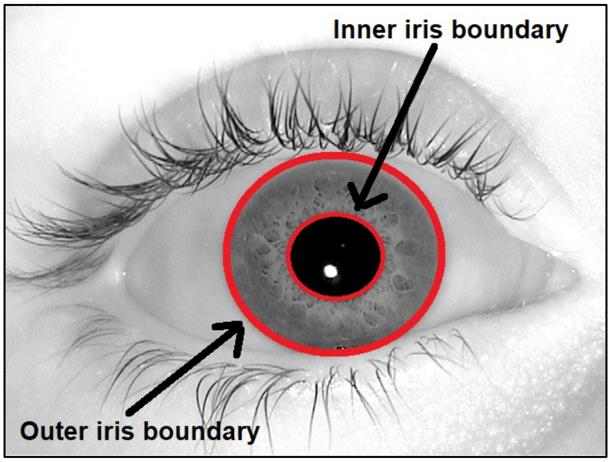


Figure (3.2): Inner and outer iris boundaries

Then normalization is applied to the selected area to reduce its size and to extract its features accurately.

3.3.1 Iris Segmentation

In this work, the goal is to detect the required area of the eye which is the iris area, without the pupil area. Since the images in the dataset are for eye images, we noticed that there is a common feature that could be based on to detect the iris region. Each image has central and almost cycle shape in the middle of the image, with a darker colour than the area around it which represents the pupil area. The properly locate and distinguish the iris pattern in the input eye image is a key stage of the biometric recognition process. By applying two stages, an automatic algorithm of segmentation would localize the iris region from an eye image. The iris is located in two stages: in the first stage is the detection of inner iris boundary (outer boundary of pupil). And the second stage, is the detection of the outer iris boundary. The resulted image will contain the region of interest (ROI), which is the iris of the eye only. These steps are shown in Figure (3.3).

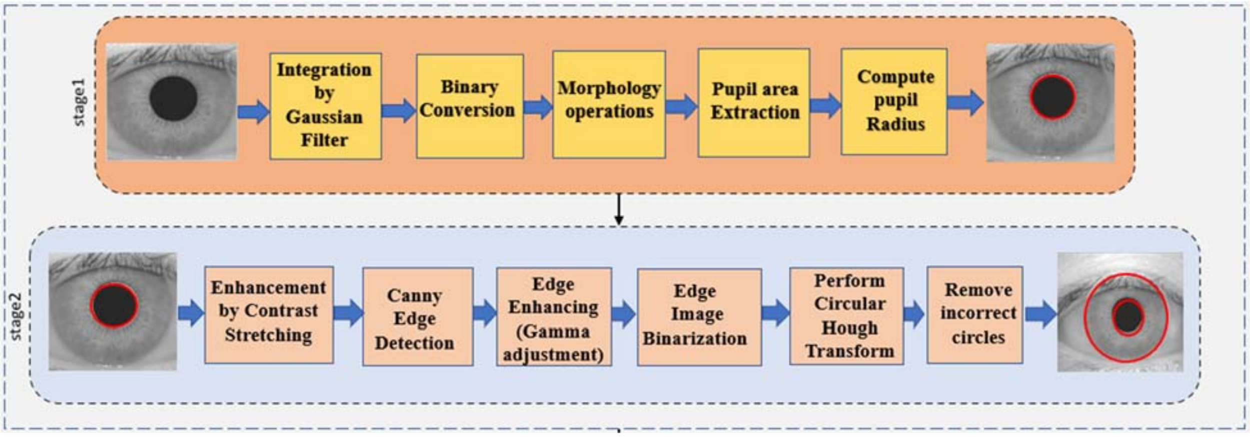


Figure (3.3): Steps of Iris Segmentation.

Stage1: Inner Iris Boundary Detection

The behavior of image intensity in both the pupil and iris of the sections of the eye is taken into account in order to determine the inner circle of the iris area which is the pupil area.

The value of the total intensity in the pupil area of the complete eye image is smaller than it is in other eye locations. Aside from that, the pupil is the largest linked and densely packed black area in the eye image. As a result, the processes in Algorithm (3.1) were used to obtain the benefits of these qualities.

```
Algorithm (3.1): Pupil Detection steps
Input: O-Img    Eye image
Output:
  $Y_p$ // coordinate of pupil center along horizontal
  xp // coordinate of pupil center along vertical
  $R_p$ // radius of pupil region
Begin
    Step1: Read eye image (O-Img).
    Step2: Apply Gaussian Smoothing Filter on (O-Img) to obtain smoothed image
              (G-Img) using equation (2.3).
    Step3: Convert (G-Img) to binary based histogram thresholding according to
              equations (3.1) and (3.2).
    Step4: Apply Morphology Operations on the image that resulted from step3
    Step5: Extract the pupil area by using equation (2.9), and compute the
             centroid of this area ($y_p$, $x_p$) by using equation (3.3).
    Step6: Compute pupil radius $R_p$ by using equations (3.4), (3.5), and (3.6).
    Step7: Return pupil center $y_p$, $x_p$ and pupil radius $R_{p.}$
End.
```

Step1: Integration by Gaussian Filter

In this step, which is the first step at this stage. The elimination of the effect of artifacts in the eye image and improve it as desired is necessary. Therefore, smoothing is applied to the entire eye image using a Gaussian filter with mask size (3×3) according to the equation (2.3) described in chapter two.

Step2: Binary Conversion

The correct intensity value to employ as a threshold in order to binarize the image into two types of pixels: (pupil and non-pupil) should be found in this step. The challenge of determining an ideal threshold value suitable to all eye images can be regarded unreasonable due to the broad range of brightness distribution difference of eye images. Furthermore, for every threshold value, some pixels which have an intensity value lower than the threshold value and may not be part of the pupil area. To address these two issues, a threshold value is generated using first-order statistical analysis based on the intensity distribution, and steps of cleaning are used to the resulting binary image to eliminate from non-pupil pixels. The image histogram is divided into five bins, since the pupil pixels have the lowest value near or equal to zero. The gray level *G* that corresponds to average of histogram bins (1 and 2) will be used as threshold as in conditional formula (3.1).

$$G = (img_{Bin(1)} + img_{Bin(2)})/2 \qquad \text{..... (3.1)}$$

Where:

img_{Bin} *(i)*: The histogram frequencies at gray level *i*.

In general, all intensity values below (T) are changed to 1 (consider as object), and all intensity values above or equal to T are changed to 0 (consider as background), that is:

$$Bim(x,y) = \begin{cases} 1 \ \ if \ \ I(x,y) \leq T \\ 0 \quad otherwise \end{cases} \qquad \text{...... (3.2)}$$

Where:

I (x, y): The intensity value at location (x, y).

Bim (x, y): the pixel value that has been converted.

Step3: Morphology Operation

Eye image contains in the pupil region white points. In CASIA-V4, the pupil region contains eight roughly white dots randomly located inside it. The main white segment is represented by the backdrop area (i.e. the area surrounding the pupil region), and the other white spots are reflection locations inside the pupil (which should convert to black points). To detect the presence of these reflection points, the closing morphology process with kernel (5×5) is used to the resulting (pupil/non-pupil) binary images according to the equation (2.7).

The morphological opening procedure (equation (2.6)) is used to reduce the effect of pores/gaps appearing in the binary image. Using a window with a size of (11×11), the opening process is carried out.

Step4: Pupil Area Extraction

To collect the pupil region, the connected components in 2-D binary image are extracted by using 8-neighboors. Then the area of each connected component is computed according to the area calculation equation (equation 2.9). Then the component with largest area represents pupil region. The center of the pupil (X_p,Y_p) is calculated by averaging the coordinates of the points in the pupil area using the equation:

$$x_p = \frac{1}{N}\sum_{i=1}^{N} x_i \quad , \quad y_p = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad (3.3)$$

Where, N is the number of collected points in pupil regions.

x_i is the ith row

y_i is the ith column

Step5: Compute Pupil Radius

To calculate the radius of the pupil, move in four directions (up, right, down, and left) from the specified point (x_p,y_p), which is the center point we defined in the previous step. For each direction, the first pixel of the background is found. Let x_l be the first background pixel detected on the left side of the line ($y=y_p$) during the horizontal scan, and x_r be the first pixel of the background found on the right side of the same horizontal scan. The horizontal radius, R_h, is then calculated as follows:

$$R_h = \frac{1}{2}(x_r - x_{l)} \qquad \text{……. (3.4)}$$

When a vertical scan is done down the column ($x=x_p$), let x_b, x_t be the first met background pixels to the bottom and top sides, respectively. R_v is the vertically assessed radius, and it's calculated like this:

$$R_v = \frac{1}{2}(x_b - x_t) \qquad \text{……. (3.5)}$$

Then, the pupil radius Rp can be calculated as follows:

$$R_p = \frac{1}{2}(R_h + R_v) \qquad \text{……. (3.6)}$$

Figure (3.4) shows how the proposed method steps on detecting the pupil area.

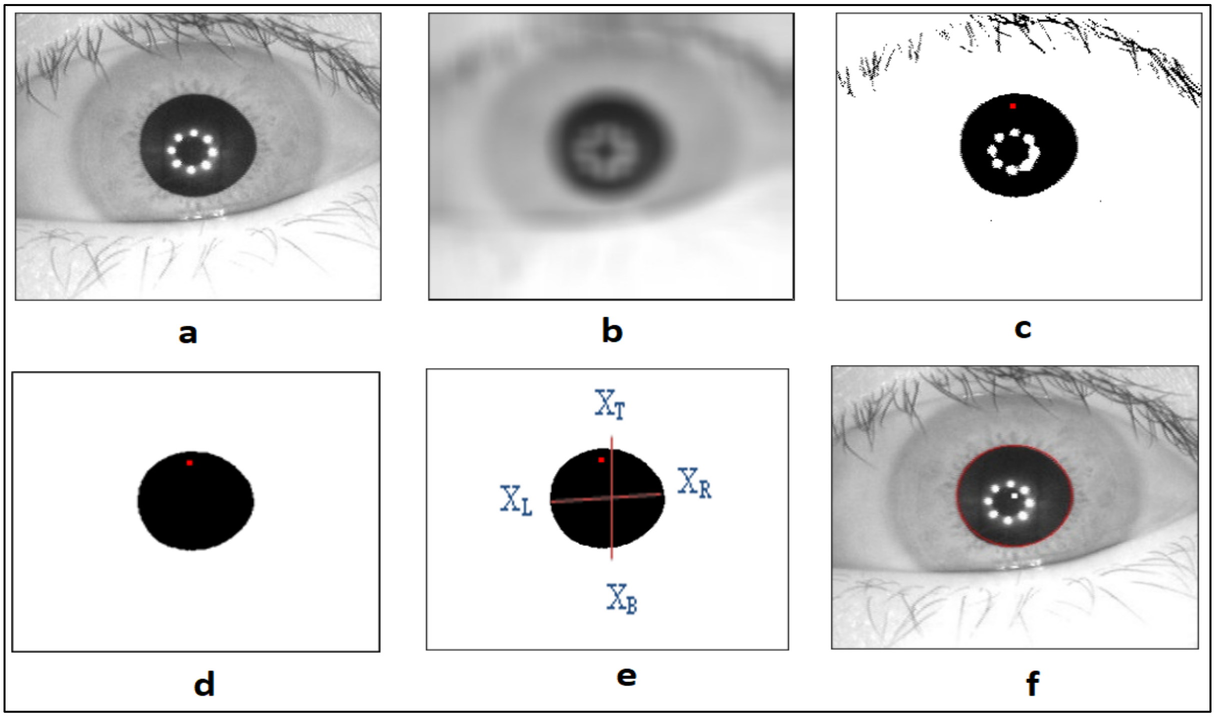


Figure (3.4): Results of pupil segmentation for CASIA-V4 dataset (a)Input image (Original) (b)Image after smoothing (c) Convert to binary image (d)Applying the morphology operation and reflection point removal results (e) The pupil region with four directions (f) Detected pupil region

Stage2: Outer iris boundary detection

Segmentation of outer boundary of iris image was achieved through using the Circular Hough Transform (HT). In this stage the iris is detected in eye image using steps depicted in Algorithm (3.2).

Algorithm *(3.2): Outer Iris Boundary detection*

```
Input: O-Img       original eye image.
       yp, xp // coordinates of pupil region center
       Rp // radius of pupil
Output: Outer Iris boundary
Begin
   Step1: Read eye image (O-Img).
   Step2: Apply enhancement process on (O-Img) to obtain enhanced image (E-Img)
          using equation (3.6).
   Step3: Apply edge detection process on (E-Img) by using Canny edge detector.
   Step4: Apply Gamma adjustment on the image (E-Img) that result from step 3 to
          enhance iris edge and obtain (G-Img) by equation (2.2).
```

```
    Step5: Convert edge image (G-Img) into binary image.
    Step6: Apply Circular Hough Transform (CHT) on result of step (5) to detect the
            circle around iris region using Equation (2.1).
    Step7: Obtain iris circle by removing incorrect circular.
    Step8: Detect Outer Iris boundary
End.
```

Step1: Eye Image Enhancement

Two interface region (pupil/iris) and (iris/sclera) make up the eye image. To make the iris border more visible, firstly, a contrast stretching is used based mapping approach on the original eye image according to the equation:

$$E - Img(X,Y) = \begin{cases} 0 & I(x,y) \leq Low \\ 255 \times \frac{I(x,y)-Low}{High-Low} & if\ Low < I(x,y) < High \\ 255 & I(x,y) \geq High \end{cases} \quad \ldots (3.6)$$

Where $E - Img(X,Y)$ represent the enhance image, Low, High is the lowest and highest gray -levels in image, respectively.

Step2: Eye Image Edge Detection

In order to detect the iris of the eye, the first step will be to apply canny edge detection on the image. This will convert the image into lines that represent the edges of the image. Canny edge detection is the process of finding the intensity gradient of the image first. Then a threshold value is applied to suppress the small and irrelevant parts of the image, for this edge map, the hysteresis threshold will be applied. With this operation, the values of the image above and below a threshold value will be omitted. The goal of this step is to ensure that the important edges are kept together, with no multiple edge parts.

Step3: Edge Enhancing Using Gamma adjustment

The edges obtained from previous step are not very clear therefore gamma adjustment with (γ=1.9) is applied to enhance the contrast of images according to equation (2.2). The value of γ can be set with in the range [1-2] in order to reduce the illumination. It is also known that the value of the gamma when it is greater than one gives a somewhat dark image, so it is used here to solve the problems of excessive lighting in the image.

Step4: Edge Image Binarization

In this step, the gamma enhanced image is converting to binary edge image. Hysteresis thresholding method is used which needs two threshold T1, and T2. All pixels with values greater than T1 are considered edges. All pixels with values over threshold T2 that are next to points that have been defined as edges are also marked as edges. Eight connectivity is utilised.

Step5: Iris Boundary Detection (Circular Hough Transform)

In the edge image, there are more circular edges along the iris boundary, and a circular Hough transform will be used. In Hough transform, the goal is to find the features that match the predefined shape. Since in Hough transform, only regular and predefined shapes could be detected, such as lines and circles. Even if the circles are not clear and incomplete, Hough Transform could identify them. As explained in detail in chapter two. Figure (3.5) shows how the proposed method steps on detecting the outer Iris boundary. The circle that has maximum in the Hough space will be taken and remove all other incorrect circles.

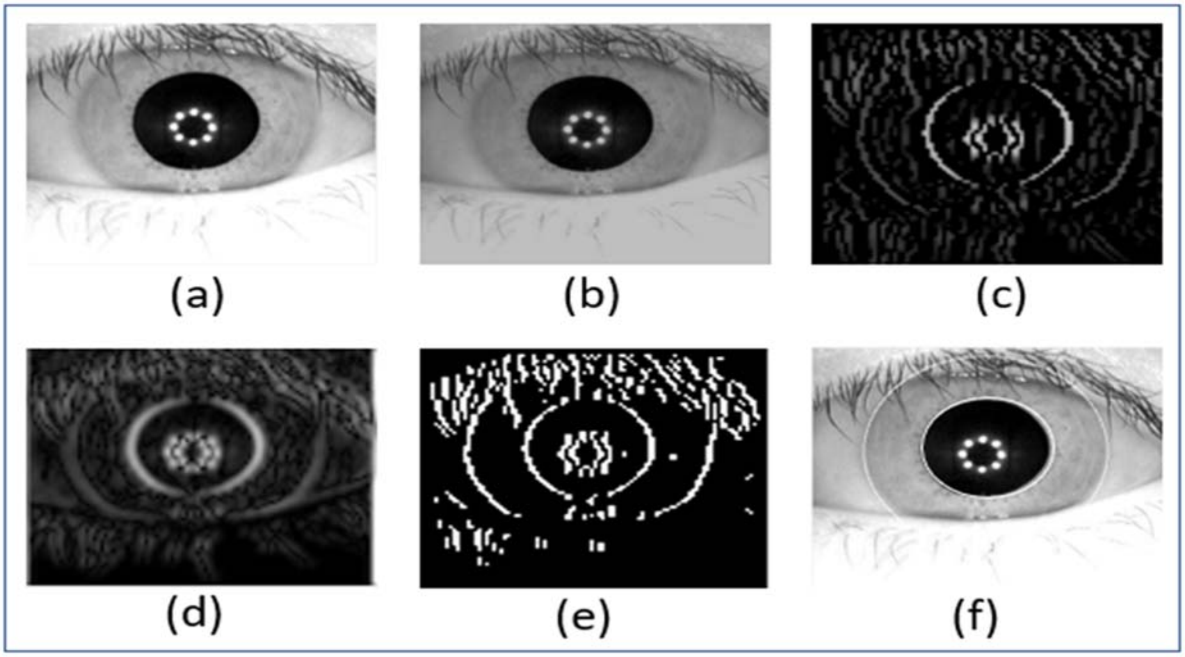


Figure (3.5): Resultant images of iris boundary (a) original image (b) enhanced image (c) image after canny edge detection (d) image after gamma correction (e) binarized edge image (f) resultant image with iris boundary

3.3.2 Iris Normalization

In this point, the segmented iris images have been converted from Cartesian coordinates to polar coordinates according to Daugman's rubber sheet model which are detailed in chapter two. Figure (3.6) shows the results of the pre-processing steps including iris normalization process on a sample of eye image.

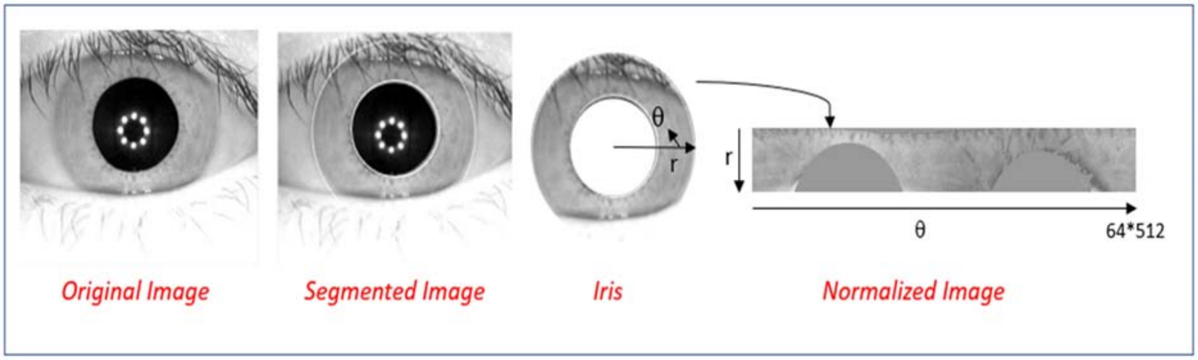


Figure (3.6): The results of the preprocessing steps

3.4 Features Extraction Using CNN

Feature extraction is the process of extracting highly discriminative features from the iris image for correct authentication purposes. In this work, a CNN network model is built and train. This modal consisting of a number of layers and hyper-parameters. The design and training of the model were adjusted to obtain high accuracy for classifying iris data. A Deep Learning technique is employed using repeating neuron blocks in the form of convolutional, pooling, and fully connected layers in order to extract the features from an acquired normalized iris image as shown in Figure (3.7).

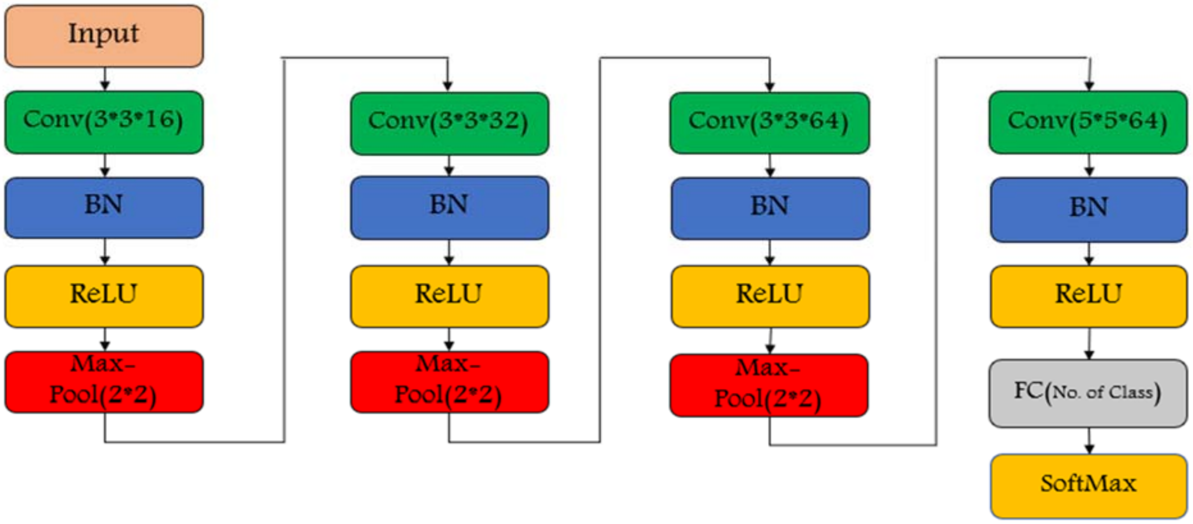


Figure (3.7): Architecture of proposed CNN

3.4.1 CNN Design

Proposed CNN network layers are:

I. Convolutional layers

The proposed model consists of four layers of convolution that define certain features or patterns in the input image. The outputs of these layers are stored in matrices called Feature Map or Activation Map, as shown in column "Activation" of the Table (3.1). Here, a number of filters of different sizes, or what is known as a kernel, are used to warp the whole image and extract the features, as shown in columns “filter size” and “#filter” of the Table (3.1). It is also used with each convolutional layer a batch normalization layer. It can be used as regularization to prevent the model from being overfitted as well as to normalize the output of the earlier layers. ReLU function is used as an activation function after each convolutional layer.

II. Max pooling layers

Max pooling is to reduce the size of the feature map by setting the largest value in each window, which not only reduces the number of required computations, but also prevents falling into the problem of Overfitting. 2×2 window size is used, and Stride 2 for all pooling layers.

III. Fully connected layer

In the multi-layer perceptron, this layer is the final one in which all of the nodes from the layer before are entirely connected to the neurons. The final classification step is completed in this layer. SoftMax activation function is used.

Table (3.1): Details of various layers of the proposed CNN network

Layer Type	*Activations*	*Filter\Window Size*	*#Filter*	*Stride*	*Padding*	*Learnable weight*
Input Image	*64*512*1 Height*Width *Channel*					
Convolution1+BN+ReLU	*64*512*16*	*3*3*	*16*	*3*	*1*	*160*
Max Pooling	*32*256*	*2*2*	*---*	*2*	*0*	
Convolution2+BN+ReLU	*32*256*32*	*3*3*	*32*	*3*	*1*	*4,640*
Max Pooling	*16*128*	*2*2*	*---*	*2*	*0*	
Convolution3+BN+Relu	*16*128*64*	*3*3*	*64*	*3*	*1*	*18,496*
Max Pooling	*8*64*	*2*2*	*---*	*2*	*0*	
Convolution4+BN+ReLU	*8*64*64*	*5*5*	*64*	*5*	*1*	*102,464*
Fully Connected +SoftMax	*No. of Class*					*4,761,800*
Sum of Learnable Weight						*4,882,920*

3.4.2 Training Hyper-parameters

During CNN training, a set of hyper-parameters that have a major role in increasing network accuracy and training speed are set as follows:

I. **Dataset split:** This step is done after the pre-processing the eye image to obtain the iris only as the data is split into 70% training and 30 % testing.

II. **Loss function:** In this model, the sparse multiclass cross entropy loss is used according to the equation (2.21) because we are dealing with a multi-class classification problem, in addition to the fact that memory needs to complete the training faster.

III. **Optimizer:** stochastic gradient descent with momentum (SGDM) described in the previous chapter is used. The momentum value 0.9 and the learning rate is 0.01.

IV. **Batch size**: the training data is divided into batches to be trained. The mini-batch size used in this model is 64.

V. **Epochs**: the number of epochs used to train CASIA-Iris V1 Dataset is 50, while CASIA-Iris v4 Dataset needed 100 epochs to get the required accuracy.

VI. **Early stopping**: in designing this network, the early stopping technique was used to eliminate the overfitting problem where training stops if the validation accuracy decreases for the last 6 epochs.

After explaining the proposed network architecture and tuning the hyper-parameters, Figure (3.8) shows the steps of training process.

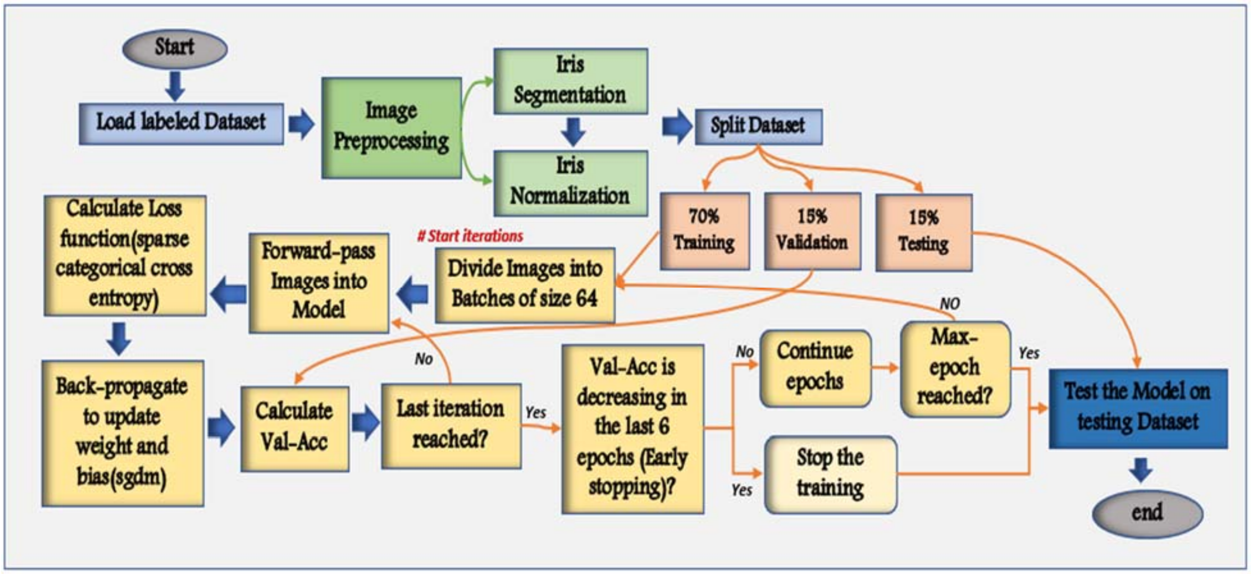

Start
Load labeled Dataset
Image Preprocessing
Iris Segmentation
Iris Normalization
Split Dataset
70% Training
15% Validation
15% Testing
# Start iterations
Divide Images into Batches of size 64
Forward-pass Images into Model
Calculate Loss function(sparse categorical cross entropy)
Back-propagate to update weight and bias(sgdm)
Calculate Val-Acc
Last iteration reached?
No
Yes
Val-Acc is decreasing in the last 6 epochs (Early stopping)?
No
Yes
Continue epochs
Stop the training
Max-epoch reached?
NO
Yes
Test the Model on testing Dataset
end

Figure (3.8): CNN Training and testing Process

In the training phase of CNN, The train is performed on the eye image; the train algorithm begins with preprocessing as described in Algorithm (3.3) that illustrates the steps of training CNN. The following step is to pass the normilized images on the convolutional neural network structure in the forward direction and the backward direction within a number of epochs depending on an early stopping to decrease the error between the predicted output of the CNN model and the actual label of the training sample, and adjust the weights to be used in the testing phase. The output of a convolutional neural network training algorithm is a trained set of weights and kernels for all layers of the network architecture. These trained

weights and kernels are stored for use by the network later in the testing process.

3.5 Classification

After the features of the images are extracted by using CNN model. Now these features should be classified by using classification algorithms. In this work, three approaches of classifications are used as shown in Figure (3.9).

3.5.1 ANN + SoftMax Classifier

Algorithm (3.3): Convolutional Neural Network (CNN) training	
Input	*$Train_{set}$: normalized iris images in the Training sample database* *E: the maximum number of epochs* *CNN: the design of proposed CNN*
Output	*Features of normalized iris images. #trained sets of kernels values for all "convolutional layers",and trained sets of weights for all "fully connected layers".*

```
Begin
  Step1: For each epoch in E
  Step2: Divide the images in $Train_{set}$ into batches of size 64
  Step3: Pass forward the $Train_{set}$through the CNN model
  Step4: Calculate the Cost Function by using Categorical Cross Entropy
  Step5: Update the weights by using SGDM optimizer
  Step6: Pass the $Val_{set}$ through CNN
  Step7: Calculate validation accuracy
  Step8: If (the validation accuracy is decreasing in the last 6 epochs): #Early Stopping
  Step9: Terminate the For loop.
  Step10: Go to Step1.
  Step11: Return Deep Features from Conv4 layer
End.
```

To determine the identity of the person, its eye image will pass to the proposed biometric system. Firstly, the test algorithm begins with preprocessing to determine iris region and convert it to normalized image. The next step is to pass the normalized image to the convolutional neural network structure to extract the test feature vector. This vector will classify using the trained weights in the fully connected layers and the trained kernel in the convolution layers that were stored in the training phase and applied later in the test phase.

In this classification method, the iris data of each person is classified using artificial neural networks (ANNs) represented by the fully connected (FC) layer followed by the SoftMax activation function described in the previous chapter.

In this classifier, 108 and 200 nodes are trained for both CASIA-Iris V1 and CASIA-Iris V4 respectively, depending on the number of classes in each dataset. Each value in the extracted features from the CNN is considered as input for the ANN classifier as shown in Algorithm (3.4).

```
Algorithm (3.4): Classification by ANN+ SoftMax classifier.
Input: $I_{test}$      #test iris images
Output: Prediction of Iris image class.
Begin
    Step1: Pass $I_{test}$ to CNN model.
    Step2: Classify $I_{test}$ by this model (by Fully Connected layer (ANN) and
            SoftMax Classifier).
   Step2: Return the class of test image.
   Step3: Calculate test accuracy.
 End.
```

3.5.2 Features Fusion

In this method, the training and testing processes are done as explain in Figure (3.9).

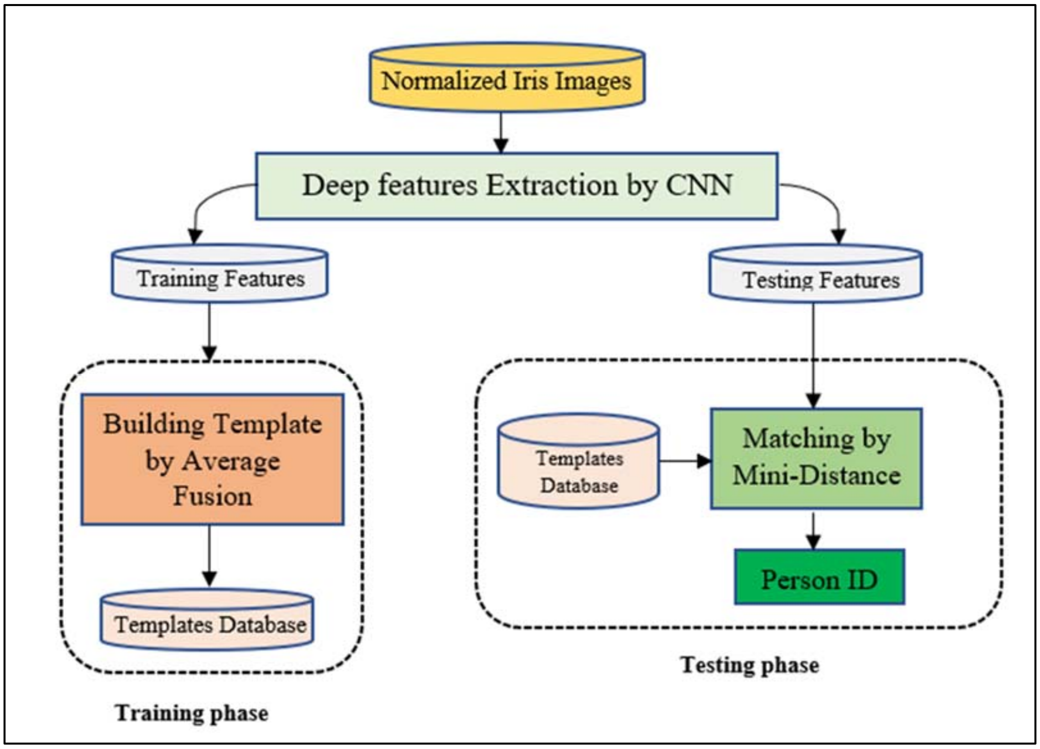


Figure (3.9): Block Diagram of training and testing processes based on Features Fusion

The features are extracted using trained CNN model. During the training phase and the construction of the template, a number of images taken in different situations of the same person are inserted into the CNN and its features are extracted, which are in the form of a vector for each image.

Hence, the features fusion technique combines various features vectors to create a single vector that represents a template for each person as shown in Figure (3.10).

Feature Fusion is a method used after extracting a feature from Convolutional Neural Network (CNN), which integrates related information extracted from a set of training images without losing any data. Feature fusion of iris recognition can considerably increase system performance and reducing the failure-to-enroll rate. The fusion process was carried out by averaging the values from each person's n-vector.

For database CASIA-Iris V1, the five features' vectors extracted from the training images were combined to build the template for each person. As for database CASIA Iris V4, seven features' vectors were combined for each person.

In the matching process, the Euclidian distance was used to determine the minimum distance between two vectors (The template from the training phase and the feature extracted from the testing phase).

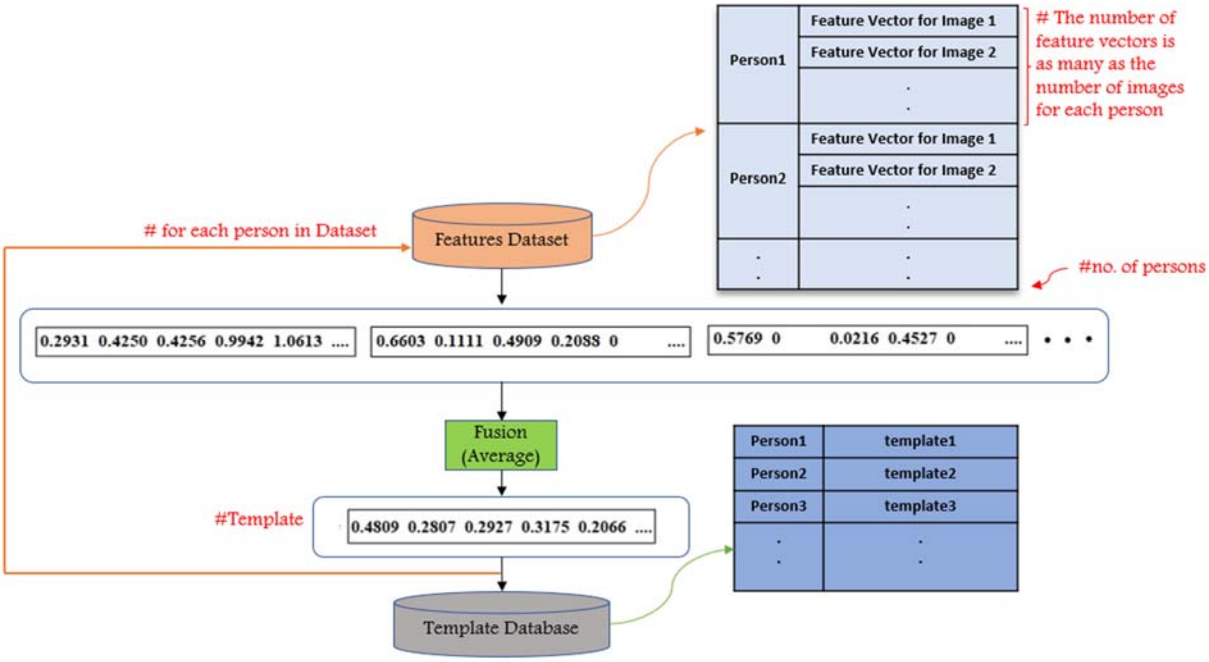


Figure (3.10): Steps of building Template for each person

Algorithm (3.5) shows the features fusion and perform the matching process by using Euclidian Distance.

```
Algorithm (3.5): Iris Features Fusion & Matching
Input: $I_i \rightarrow I_{train(70)} + I_{test(30)}$      # images in each class in database
Output: Matching Accuracy (ACC).
Begin
  Building the Template
    Step1: Pass $I_{train}$ to CNN Model to get the features $F_{Train}$.
    Step2: Build the Template for each class (person), by applying this
           Equation: $Final_{Temp} = Avg(F_{train})$.  #Avg: Average
    Step3: Store $Final_{temp}$ in Database.
    Step4: Repeat Step2 and Step3 for all Classes in database.
  Matching
    Step5: Pass $I_{test}$ to CNN Model to get the features $F_{test}$ .
           # For iris image in each class in the testing samples database.
    Step6: Match the $F_{test}$ with $Final_{temp}$ by applying Euclidian
           Distance according to the equation (2.30).
  Step7: Compute Accuracy (ACC).
End.
```

3.5.3 SVM Classifier

In this method, and after feature extraction, the SVM classifier is used to discover the associated label for each test image as shown in Algorithm (3.6). The SVM receives the features extracted from CNN model, and treats them as inputs. It tries to distinguish the patterns revealed from CNN.

Practically in MATLAB, “fitcecoc” function was used, which is one of the SVM functions that are suitable for multi-class models, because it reduces the computation time for high-dimensional data, and it trains the data more efficiently.

```
Algorithm (3.6): Classification by SVM classifier.
Input: $I_i \rightarrow I_{train(70)} + I_{test(30)}$      # images in each class in database
Output: Prediction of Iris image class.
Begin
  Training Steps
    Step1: Pass $I_{train}$ to CNN Model to get the features $F_{Train}$.
           # For iris image in each class in the training samples database.
```

```
      Step2: Use $F_{train}$ to train the SVM Classifier.
   Testing Steps
      Step3: Pass $I_{test}$ to CNN Model to get the features $F_{test}$ .
            # For iris image in each class in the testing samples database.
      Step4: Use the trained SVM Classifier to predict the label for $F_{test}$.
      Step5: Obtain the well-known label for $F_{test}$.
      Step6: Compute the average accuracy.
End.
```

Chapter Four

Experimental Results and Evaluation

Chapter Four

Experimental Results and Evaluation

4.1 Introduction

In the previous chapter, the work of the iris recognition system is described in all its details and the sequence of its operations. In this chapter, the performance results of the proposed system, which consist of three phases, will be discussed: results of the pre-processing stage, results of features extraction stage, and results of the classification stage (CNN+ SoftMax, Feature fusion, and CNN-+SVM). This chapter also include details about the iris databases used to test the results of this system in addition to the detailed outputs for each step and the evaluation of the proposed system by calculating performance measures.

4.2 Hardware and Software Requirements

The proposed Iris Recognition System operates by using a personal computer (Laptop) hp with specifications (Intel ® Core ™ i7-8565U@1.80 GHz (8 CPUs), ~2.0GHz for CPU), (8192 MB RAM) and (Windows 10 Pro-64-bit). Programmatically, this system was implemented in the MATLAB-2020 environment.

4.3 Iris Dataset Preparations

Image Acquisition is the first stage of the work of any biometric system. In this work, we used the CASIA-Iris Dataset in two versions [81]:

CASIA Iris Image Dataset V1: The Centre for Biometrics and Security Research compiled this dataset. The CASIA iris image dataset, version 1.0, contains 756 images from 108 different persons. There are seven grayscale images for each person. With a resolution of 280 x 320 pixels.

CASIA Iris Image Dataset V4 (interval class): Here used 2000 iris images from this dataset. 8-bit gray-level JPEG files are used for all iris images that were captured or produced under near-infrared illumination. Each iris image is 280 x 320 pixels and is grayscale (256 levels). We use 200 classes; each class contains 10 images.

The datasets are divided into three groups: the training data group is 70% of the total dataset (540 for CASIA-V1, 1400 for CASIA-V4 (Interval)), and the testing data group is 30% (216 for CASIA-V1, 600 for CASIA-V4 (Interval)). Table (4.1) shows the Statistics of the CASIA-Iris divide.

Table (4.1) Statistics of CASIA-Iris V1 and V4

	CASIA-Iris V1	CASIA-Iris V4	Total
Train	540	1400	1940
Test	216	600	816
Total	756	2000	2756

Each database has a set of characteristics that distinguish it from the other, and each of them has problems, as explained earlier. The following is a set of samples on each type of database used in this research (Figure (4.1) and (Figure (4.2)).

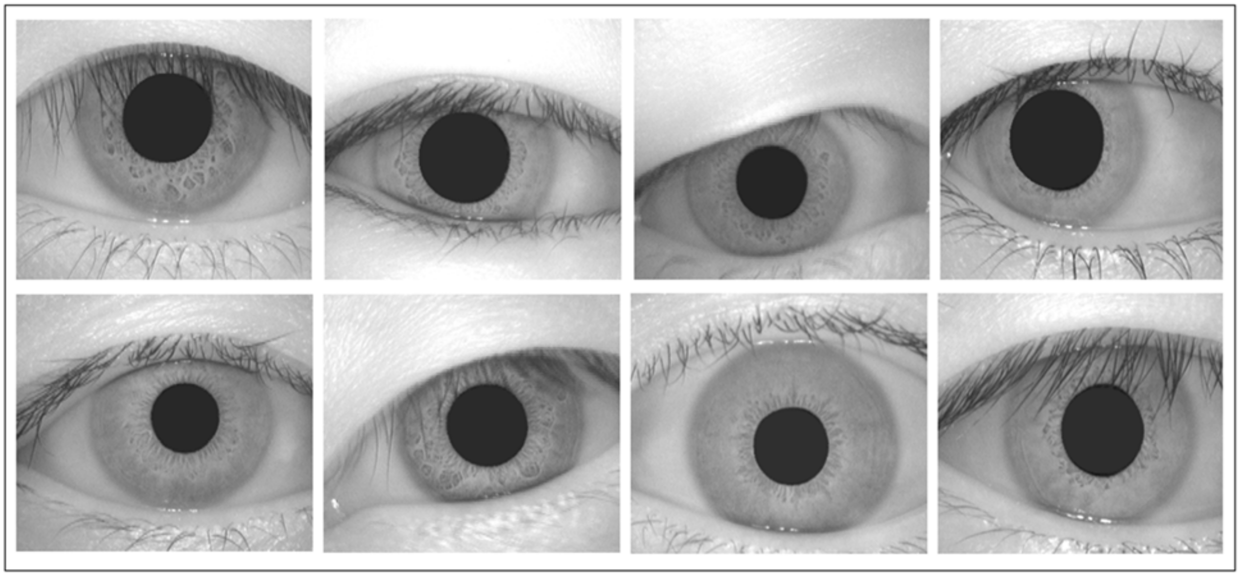


Figure (4.1): Some samples of the dataset CASIA-Iris V1 (The Original Images).

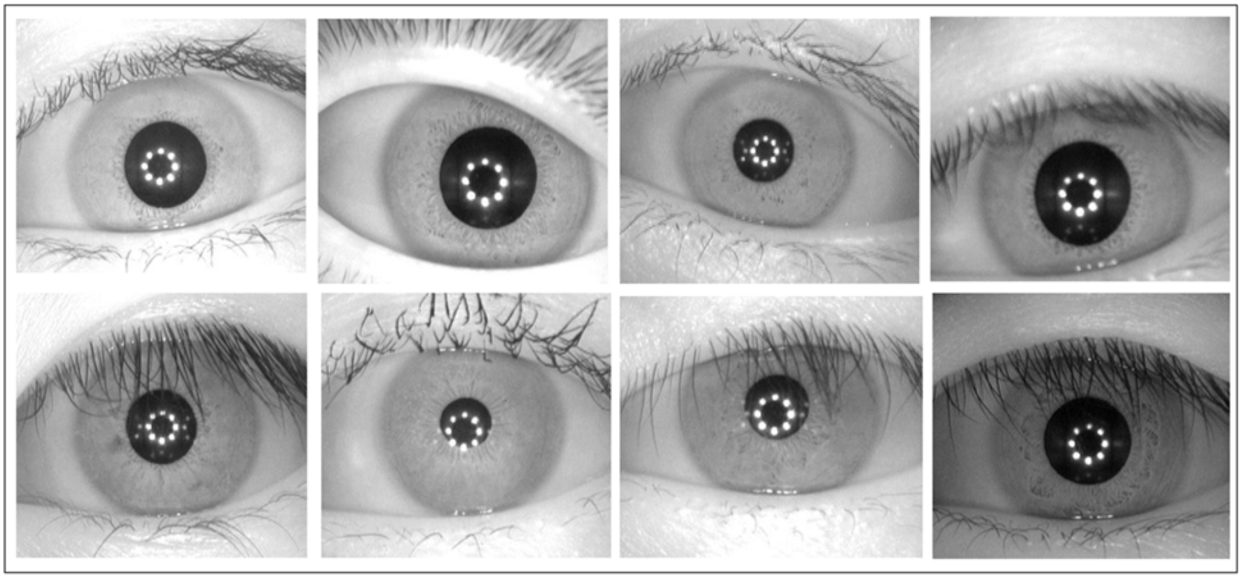


Figure (4.2): Some samples of the dataset CASIA-Iris V4 (The Original Images).

4.4 Results of the Proposed System

The results of the proposed system will be reviewed according to the three stages in which the system work, namely:

4.4.1 The Results of Pre-processing

The goal of pre-processing is to make the original image ready for feature extraction algorithms. This stage consists of several treatments and improvements that are applied to the image, they will be summarized in two stages:

A- Results of Iris Segmentation.

The process of iris segmentation, as explained in chapter three, goes through two stages:

I. Inner Iris Boundary Detection

The results of applying the steps of this stage to the databases CASIA-Iris V1 and database CASIA-Iris V4 are described in Tables (4.2) and (4.3), respectively, based on the following steps:

a) Integration by Gaussian Filter.

b) Binary Conversion.

c) Morphology Operations.

d) Pupil Area Extraction.

e) Compute Pupil Radius.

Table (4.2): Steps of the results of inner iris boundary detection (CASIA- V1)

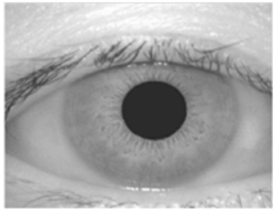
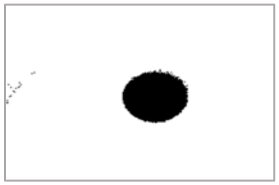
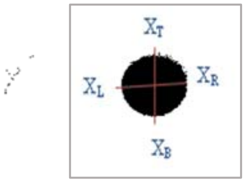
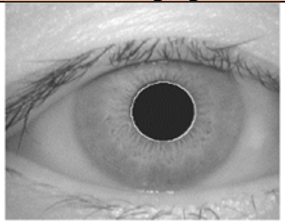
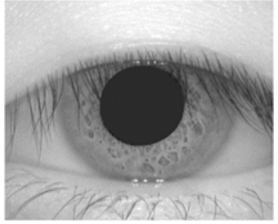
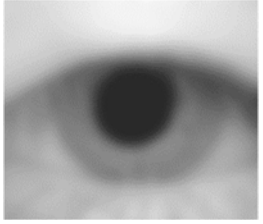
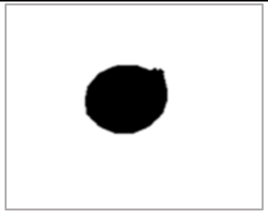
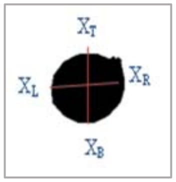
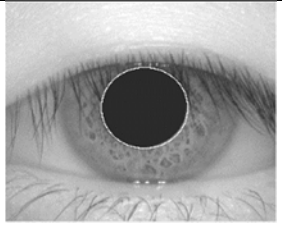
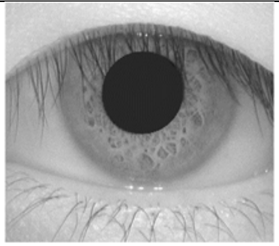
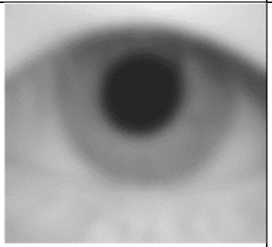
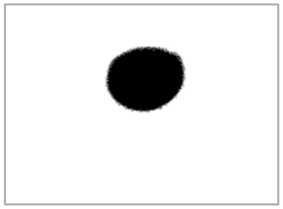
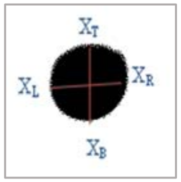
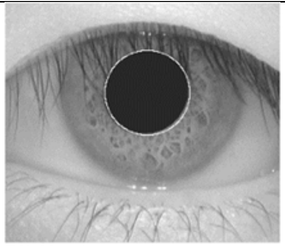
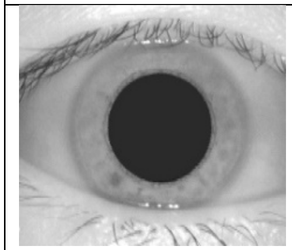
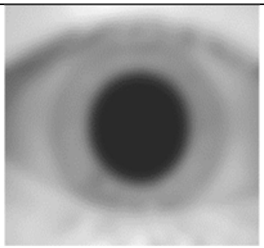
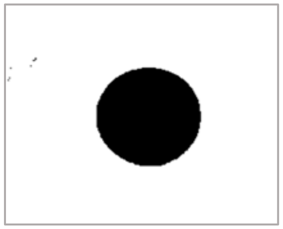
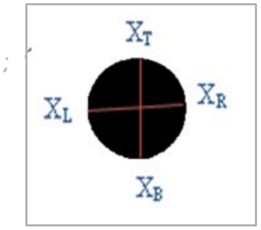
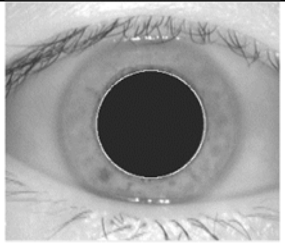
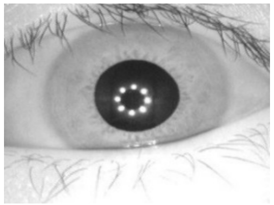
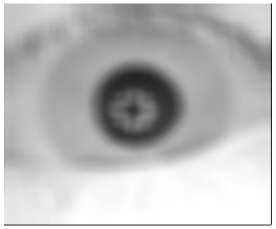
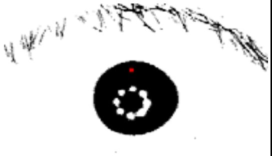
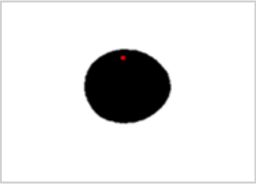
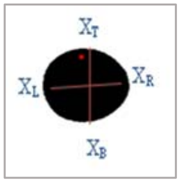
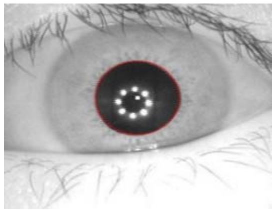
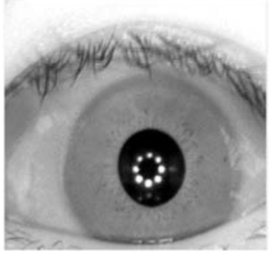
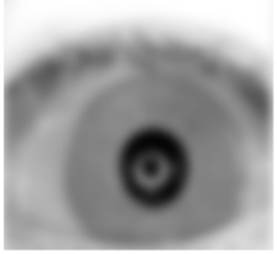
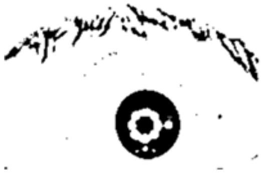
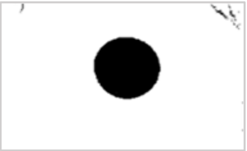
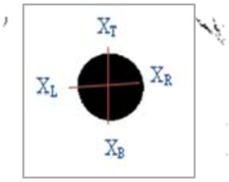
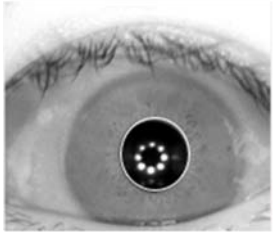
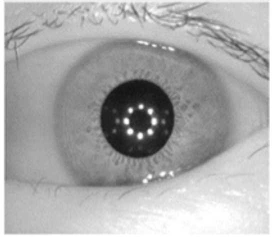
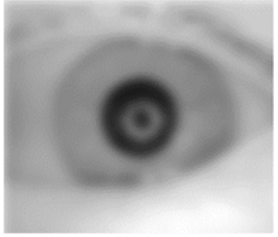
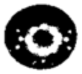
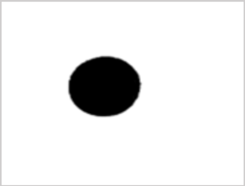
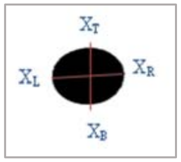
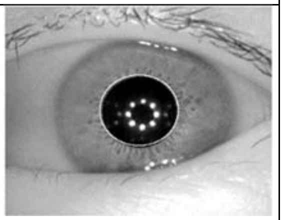
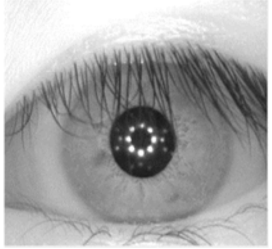
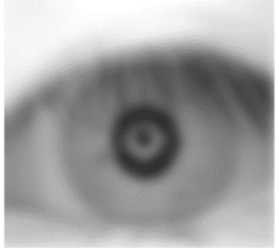
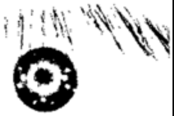
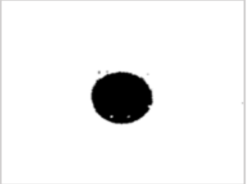
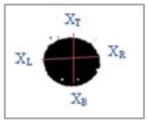
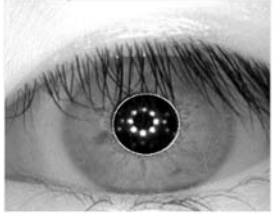
Original Image	Gaussian Filter	Binary Image	Morphology Operation	Pupil Area Extraction	Compute Radius & Detect pupil
					
					
					
					

Table (4.3): Steps of the results of inner iris boundary detection (CASIA-V4)

Original Image	Gaussian Filter	Binary Image	Morphology Operation	Pupil Area Extraction	Compute Radius & Detect pupil
					
					
					
					

II. Outer Iris Boundary Detection

The results of applying the steps of this stage to the databases CASIA-Iris V1 and database CASIA-Iris V4 are described in tables (4.4) and (4.5), respectively, based on the following steps:

a) Eye Image enhancement (Contrast Stretching).

b) Eye Image Edge detection (Canny Edge Detection).

c) Edge Enhancing (Gamma Adjustment).

d) Edge Image Binarization.

e) Iris Boundary Detection (Circular Hough Transform).

Table (4.4): Steps of the results of outer iris boundary detection (CASIA-V1)

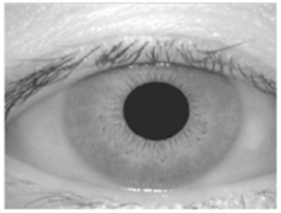
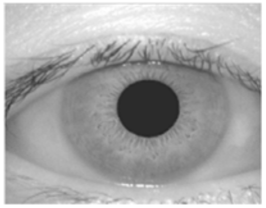
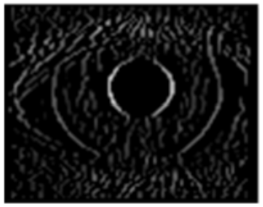
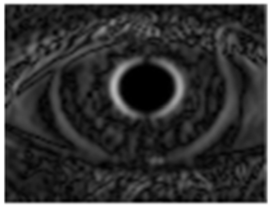
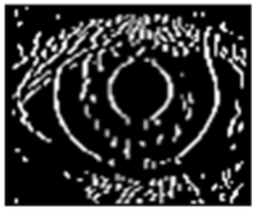
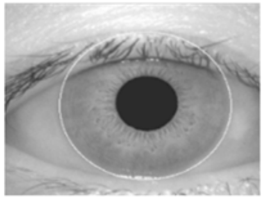
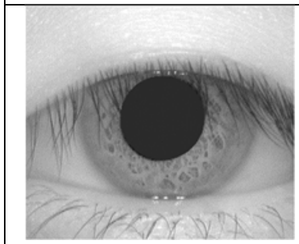
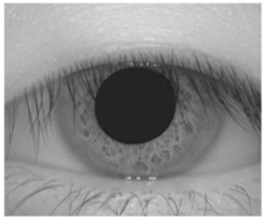
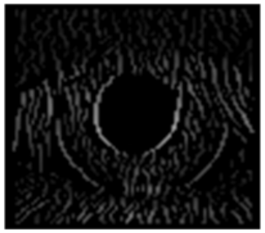
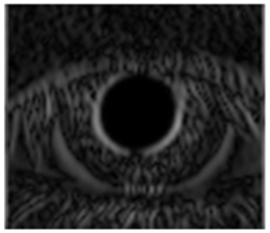
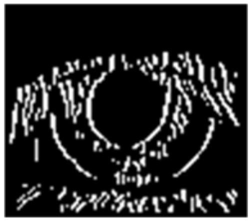
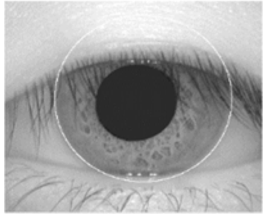
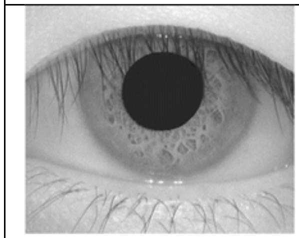
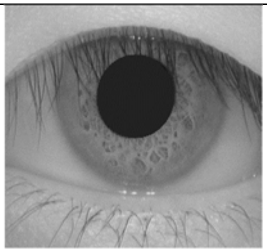
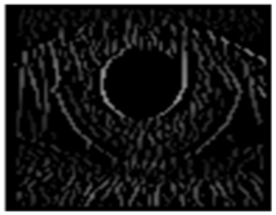
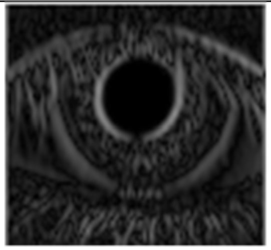
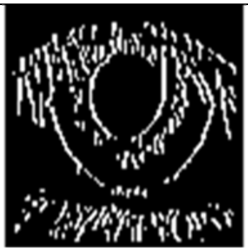
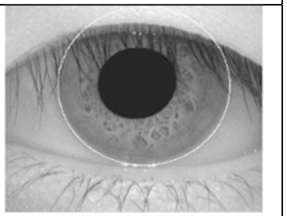
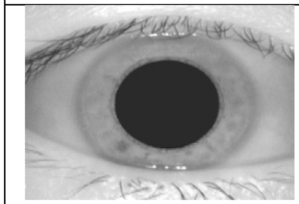
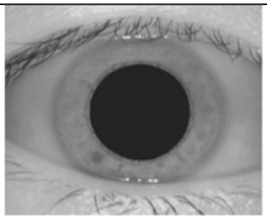
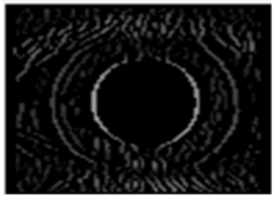
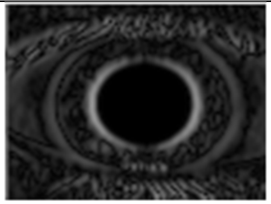
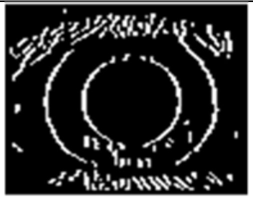
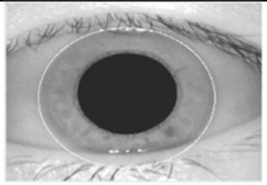
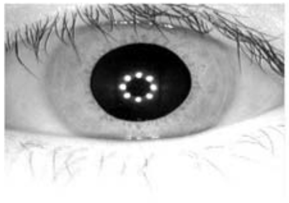
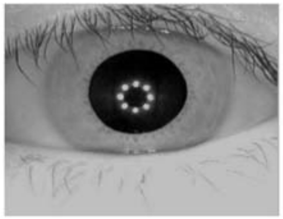
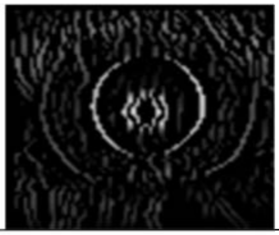
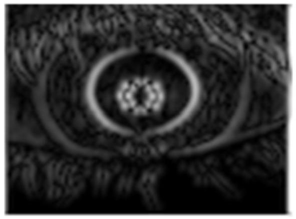
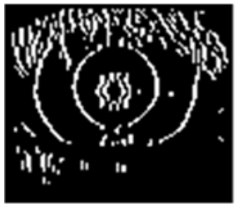
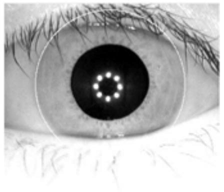
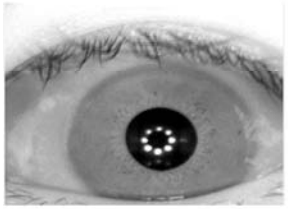
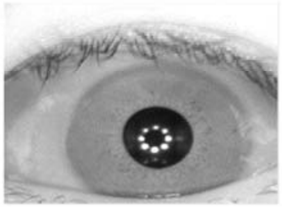
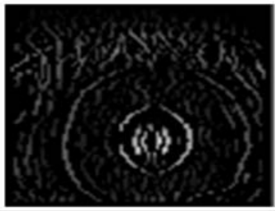
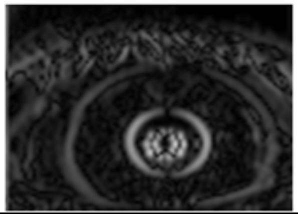
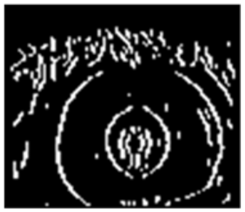
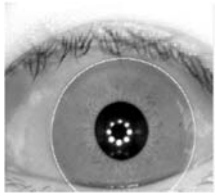
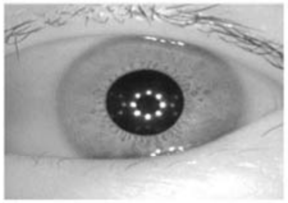
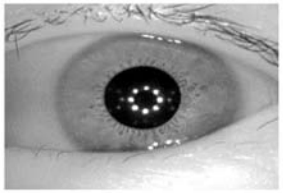
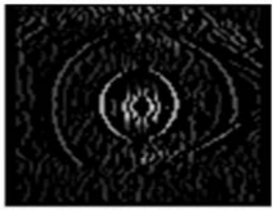
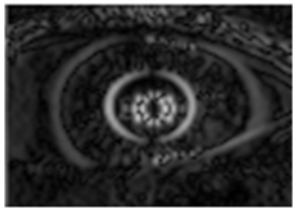
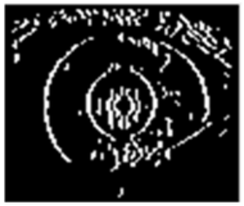
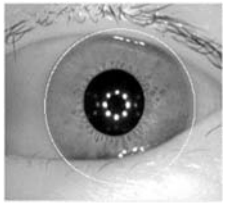
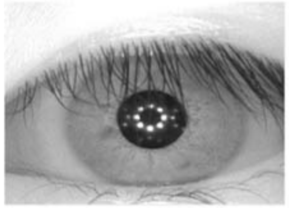
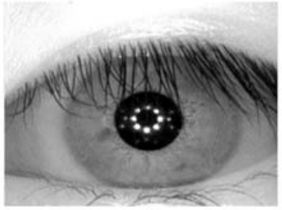
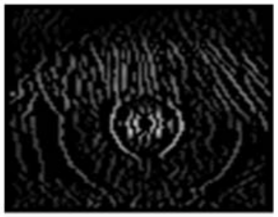
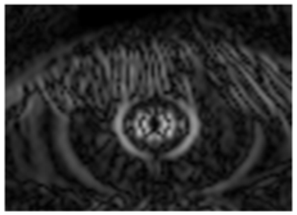
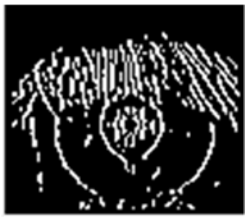
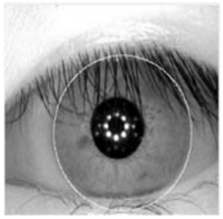
Original Images	Contract Stretching	Canny Edge Detection	Gamma Adjustment	Binarization	Circular Hough Transform
					
					
					
					

Table (4.5): Steps of the results of outer iris boundary detection (CASIA-V4)

Original Images	Contract Stretching	Canny Edge Detection	Gamma Adjustment	Binarization	Circular Hough Transform
					
					
					
					

After determining the inner and outer borders of the iris, we can observe the final results of segmentation of the entire iris region on both databases used in this work. Iris Boundary Segmentation is achieved by determining the geometrical parameters of the iris, namely the iris center and radius (inner and outer). The suggested method's results on several randomly selected images from the CASIA-V1 and CASIA-V4 datasets are shown in figures (4.3) (4.4).

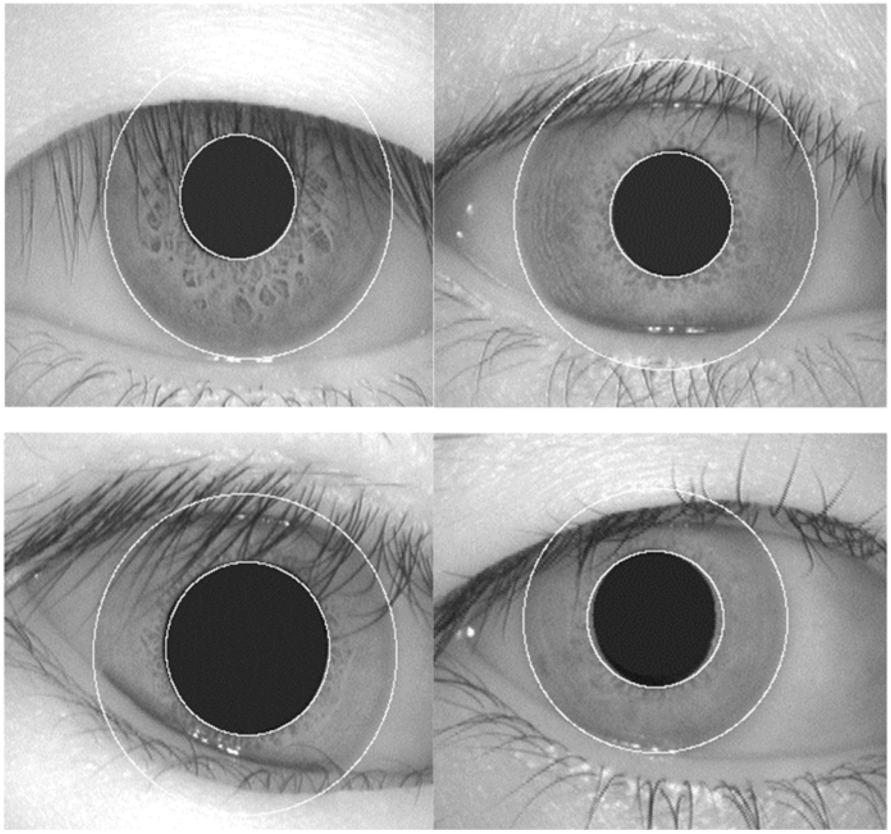


Figure (4.3): Samples of accurate Iris Segmentation results for images CASIA-Iris V1.

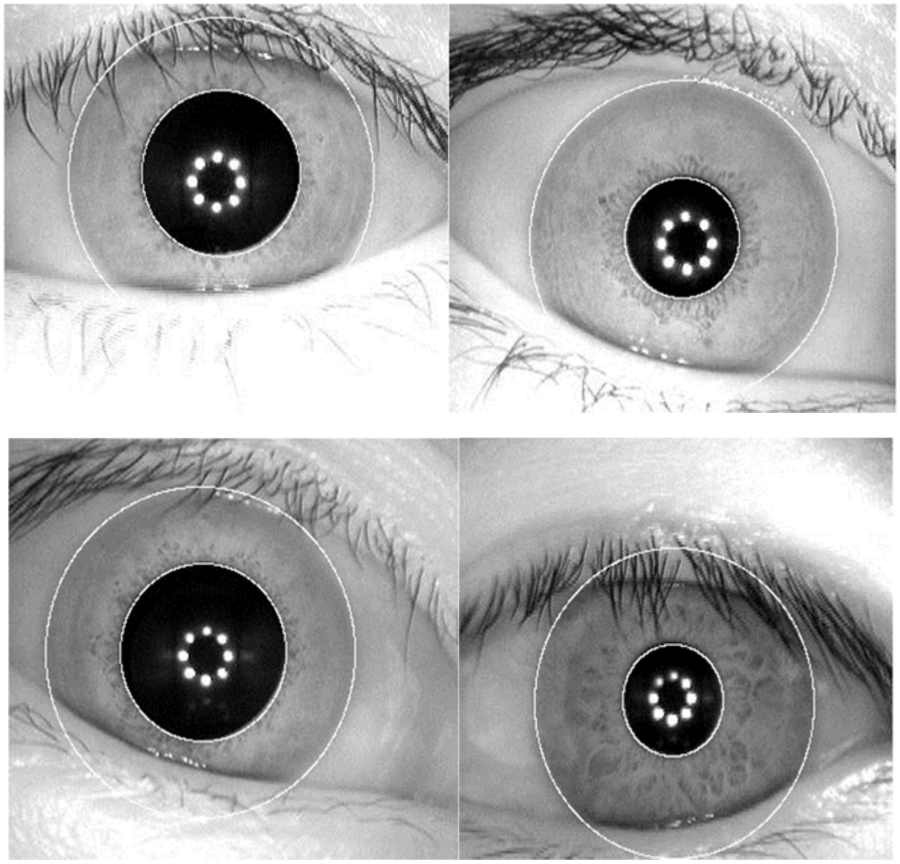


Figure (4.4): Samples of accurate Iris Segmentation results for images CASIA-Iris V4.

The results will be evaluated by calculating the accuracy of the correctly detected iris in the dataset. The accuracy is about (100%) for CASIA-Iris V1 for both inner and outer boundaries. While the accuracy for CASIA-Iris V4 is about (99.60) for the inner boundary and is about (99.16) for the outer boundary.

Table (4.6) shows the comparison of the overall accuracy with several existing methods for CASIA-Iris Dataset with its many versions.

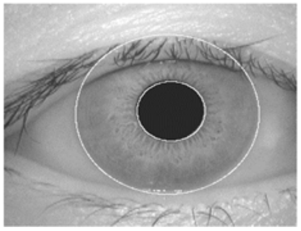
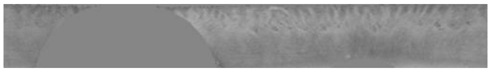
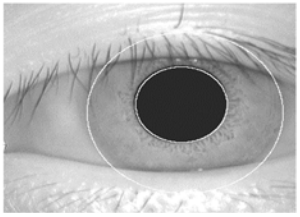
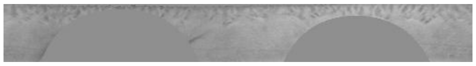
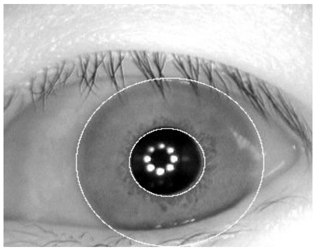
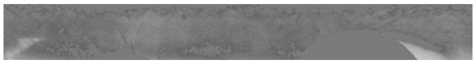
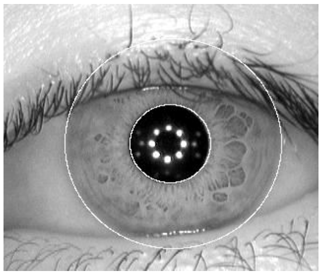
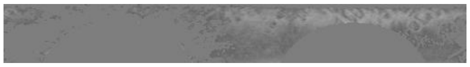
Table (4.6): Comparison between the proposed Segmentation method with the previous methods

Reference	***Method***	***A version of CASIA-Iris Dataset***	***Accuracy %***
[22]	*integro-differential operator*	*CASIA-Iris*	*91.39*
	Hough Transform		*93.06*
[27]	*Faster R-CNN*	*CASIA-V4*	*95.49*
[23]	*Morphological filter &two direction scanning*	*CASIA-V1*	*96.48*
		CASIA-V4	*95.1*
[25]	*U-Net*	*CASIA-V1*	*ranged between 96% - 97% based on network depth and BN*
[26]	*Fully dilated convolution combining U-Net (FD-UNet)*	*CASIA-V4*	*97.36*
Proposed Method	***Hough Transform+ Image Processing Techniques***	***CASIA-V1***	***100***
		CASIA-V4	***99.16***

B. Results of Iris Normalization.

It is the process of transforming the iris region from an annular shape to a fixed-dimensional rectangle. The final results are shown in table (4.7).

Table (4.7): The results of the normalization process

Segmented Image	Normalized Image
CASIA-Iris V1 Dataset	
	
	
CASIA-Iris V4 Dataset	
	
	

4.4.2 Results of Feature Extraction Stage

To extract the features from the iris pattern that was previously determined, we used a proposed model of CNN, as we have explained in detail in chapter three, where this model consists of four layers of convolution, max pooling, and fully connected layer**.**

A-CASIA-Iris V1

This model was trained on CASIA-Iris V1 which was divided into 70% training, 15% validation, and, 15% testing. The results of the training were as shown in Figure (4.5), the light blue in this figure represents the training, the dark blue represents smoothed training and, the black dotted represents validation. Figure (4.6) represents the loss function over the iteration. In this figure, the pink line represents training; the red line represents smoothed training, and the black dotted validation.

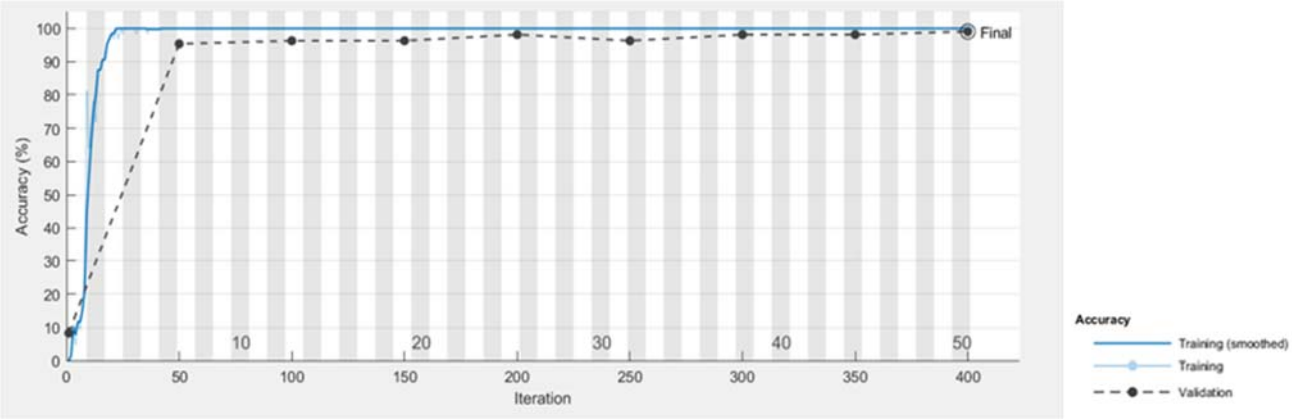


Figure (4.5): The training and validation accuracy over the iterations (CASIA-V1)

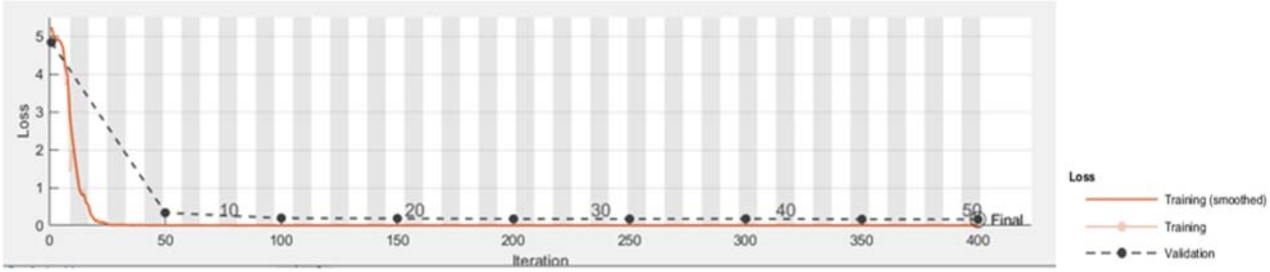
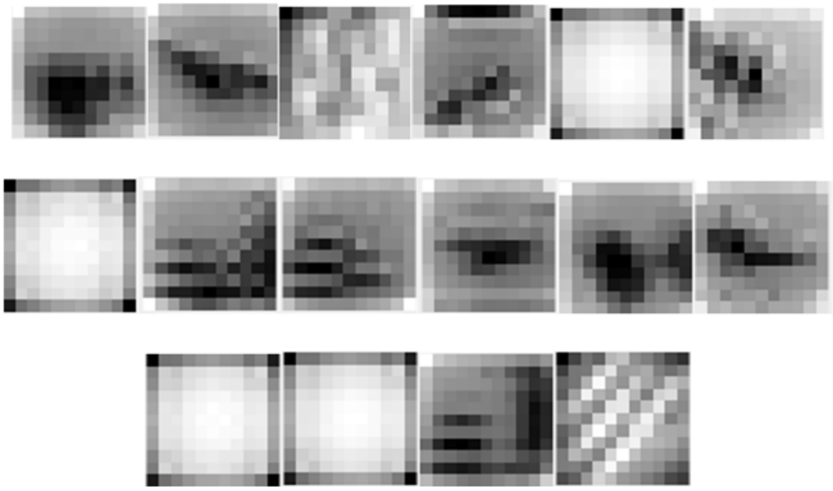
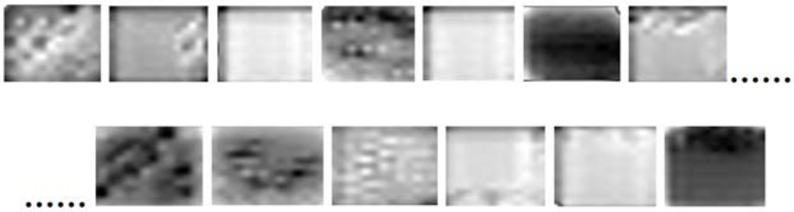


Figure (4.6): Loss Function over the iteration (CASIA-V1)

The proposed Convolutional Neural Network (CNN) outputs are shown in table (4.8), in this table, the first column contains the layer name and the feature map dimensions represented by (width*height* number), while the second column in the table displays the feature map image.

Table (4.8): the outputs of the proposed CNN Model for CASIA-V1

Layers	Feature-map images
 Conv1 ***64*512*16***	
Conv2 ***32*256*32***	
Conv 3 ***16*128*64***	
Conv4 ***8*64*64***	

B-CASIA-Iris V4

This model was also trained on CASIA-Iris V4 Dataset and the results were as shown in Figure (4.7) which shows the accuracy of the training and Figure (4.8) which represents the loss function. The features extracted from the proposed network of the dataset CASIA-Iris V4 are shown in Table (4.9).

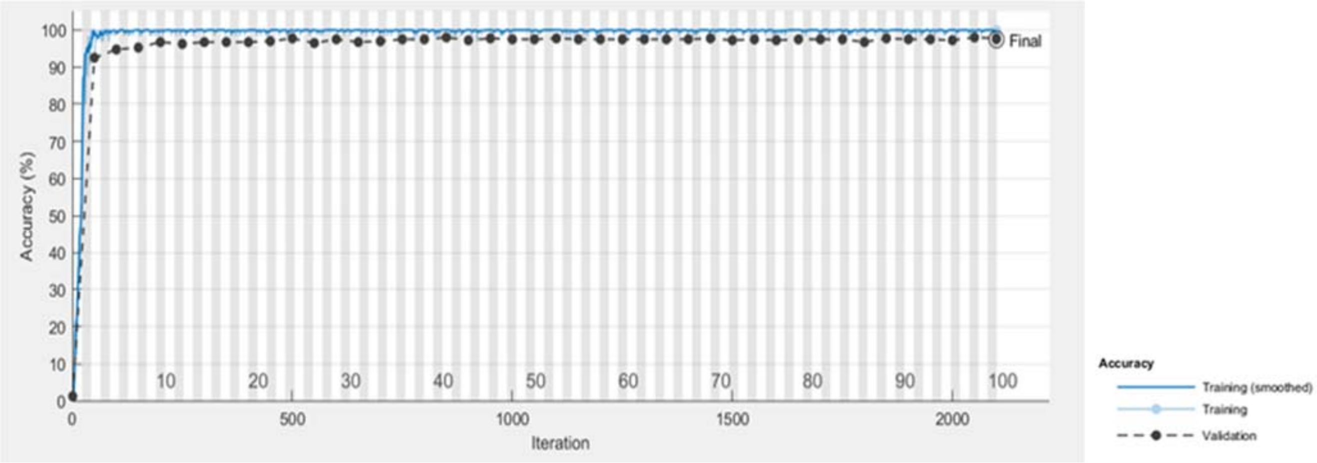


Figure (4.7): The training and validation accuracy over the iterations (CASIA-V4)

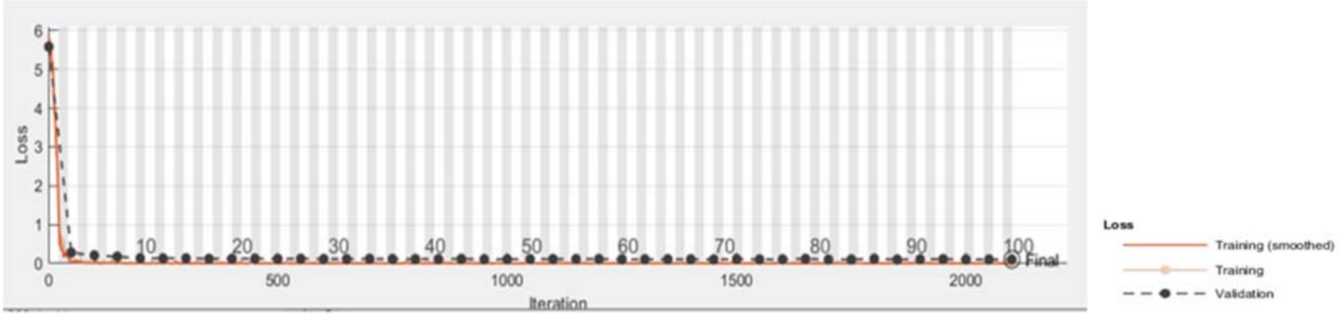
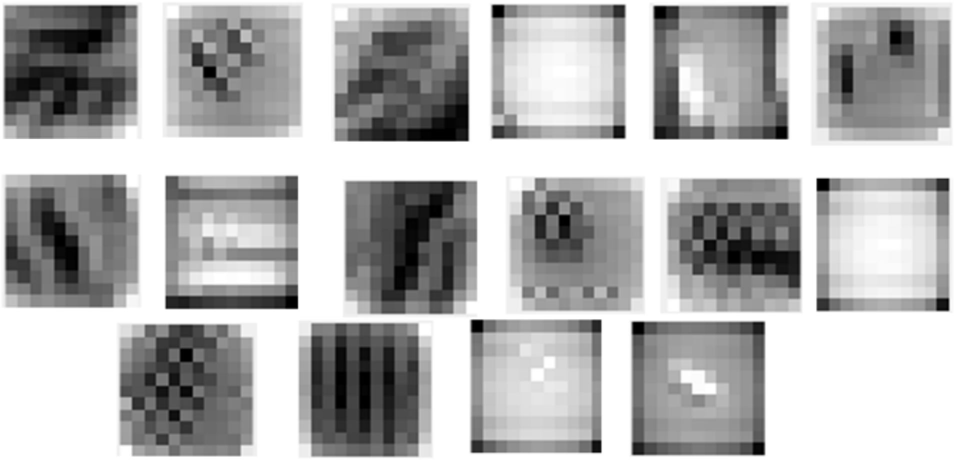
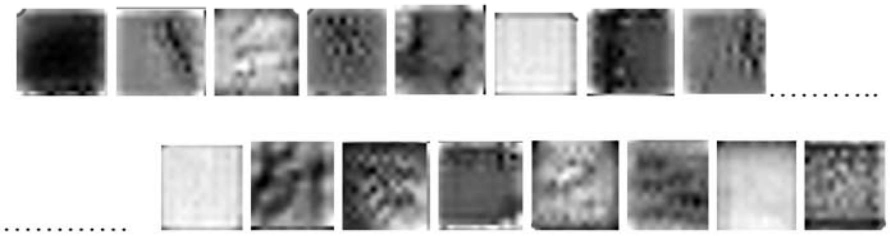


Figure (4.8): Loss Function over the iteration (CASIA-V4)

Table (4.9): The outputs of the proposed CNN Model for CASIA-V4

Layer	Feature-map images
 Conv1 ***64*512*16***	
Conv2 ***32*256*32***	
Conv 3 ***16*128*64***	
Conv4 ***8*64*64***	

4.4.3 Results of Classification Stage

I. Results of (CNN + SoftMax) Classifier

The results of classification by using the artificial neural network (ANN) represented by fully connected layer in CNN network and followed by SoftMax classifier can be displayed in the table (4.10) that explains the validation accuracy during training the model and test accuracy for CASIA-Iris V1 and CASIA-Iris V4. A set of metrics can be calculated such as precision, recall and, F1score that evaluates the efficiency of the system.

In table (4.10), includes the calculation of these measures.

Table (4.10): The classification accuracy for (CNN+ SoftMax)

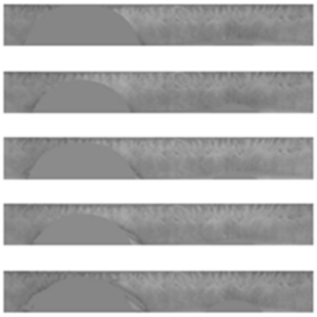
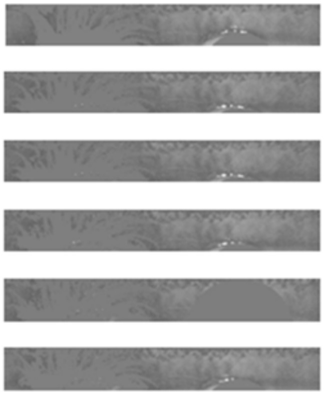
Dataset	*Train Data*	*Validation Data*	*Test Data*	*Val-Acc %*	*Test-Acc%*	*Precision*	*Recall*	*F1-Score*
CASIA-V1	70%	15%	15%	98.15	98.89	98.88	98.97	98.92
CASIA-V4	70%	15%	15%	97.50	98.5	98.4	98.66	98.52

II. Results of Feature Fusion

In the training phase, after extracting the features of the training data that make up 70% of the database, we build the template through the process of integrating the features of each person according to the algorithm (3.5) described in the third chapter.

Tables (4.11) consist of three fields representing the ‘***Dataset***’ field, which represents the databases that have been worked on, and ‘***Input Images***’ represents the number of images entered per person (5 images in CASIA Iris V1, 7 images in CASIA Iris V4), and ‘***Feature Vectors***’ represents the features extracted from the CNN. Then, by calculating the average of the values, a single vector is formed that represents the ***template***.

Table (4.11): Results of the feature fusion algorithm (CASIA-V1 & V4)

Dataset	***Input Images***	***Feature Vectors***
CASIA-Iris V1		**0, 0, 0.2299, 0.2826, 0.1093, ……** **0, 0, 0.0805, 0.0805, 0.1906, ……** **0, 0, 0.3794, 0.4567, 0.2467, ……** **0, 0, 0.0948, 0,1309, 0, 0, …….** **0.2667, 0.3739, 0.7799, 0.7100, ….**
Template		**0.0533, 0.0748, 0.3129, 0.3638, 0.2243, …**
CASIA-Iris V4		**0.2931, 0.4250, 0.4256, 0.9942, 1.0613, ...** **0.6603, 0.1111, 0.4909, 0.2088, 0, ...** **0.6754, 0.6788, 0, 0, 0.1003, ...** **0.5769, 0, 0.0216, 0.4527, 0, ...** **0.5811, 0.5119, 0.3679, 0.3040, 0.2843, ...** **0.3196, 0.2357, 0.5023, 0.0960, 0, ...** **0.2596, 0.0027, 0.2405, 0.1665, 0, ...**
Template		**0.4809, 0.2807, 0.2927, 0.3175, 0.2066, ...**

In the testing phase, the features extracted from CNN for the test data that make up 30 % of the database are matched to the template using ***Euclidean Distance.***

1. CASIA-Iris V1 Dataset

For this dataset, the test data is 30% of the dataset (216 images) which is the equivalent of (2 images) per person. Table (4.12) shows the matching of the two test images with the template for each person.

Table (4.12): Results of matching (CASIA V1)

Index	Result of Matching for Testing Images	Label (Person number)
1	**1**	**1**
2	**1**	
3	**2**	**2**
4	**2**	
5	**3**	**3**
6	**3**	
7	**4**	**4**
8	**4**	
9	**5**	**5**
10	**5**	
11	**6**	**6**
12	**6**	
13	**7**	**7**
14	**7**	
. . .	. . .	. . .
215	**108**	**108**
216	**108**	

2. CASIA-Iris V4 Dataset

For this dataset, the test data is 30% of the dataset (600 images) which is the equivalent of (3 images) per person. Table (4.13) shows the matching of the three test images with the template for each person.

Table (4.13): Results of matching (CASIA V4)

Index	Result of Matching for Testing Images	Label (Person number)
1	**1**	**1**
2	**1**	
3	**1**	
4	**2**	**2**
5	**2**	
6	**2**	
7	**3**	**3**
8	**3**	
9	**3**	
10	**4**	**4**
11	**4**	
12	**4**	
. . .	. . .	. . .
595	**199**	**199**
596	**199**	
597	**199**	
598	**200**	**200**
599	**200**	
600	**200**	

3. Performance Evaluation

To evaluate the performance of this method, we calculate the accuracy by calculating the number of images whose matches were correct over the total number of images. Table (4.14) represents the accuracy of the system. This table consists of five fields: ‘Dataset’ indicates which databases have been worked on, ‘Test Data’ percentage of test data from total databases, ‘No. of images’ total test images, ‘No. of images for each person’ number of test images per person, and ‘Accuracy’ system accuracy for each database.

Table (4.14): The accuracy of the feature fusion method

Dataset	***Test data***		***No of images***	***No. of images for each person***	***Accuracy%***
CASIA-V1	30%		216	2	96.29
CASIA-V4	30%		600	3	95.83

III. Results of (CNN+ SVM) Classifier

The features are extracted from the last layer of the convolutional neural network (CNN), which is the ***Conv4*** fed into the support vector machine (SVM). Table (4.15) also show the calculation of system performance metrics on both databases.

Table (4.15): The classification accuracy for (CNN+ SVM)

Dataset	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CASIA-Iris V1	**99.54**	**99.53**	**99.69**	**99.61**
CASIA-Iris V4	**98.83**	**98.83**	**99.22**	**99.02**

4.5 Comparison of Proposed System and Previous Methods

Table (4.16), (4.17) and (4.18) contain a comparison of various Iris recognition methods with the proposed system. It should be noted that the proposed system outperformed the other existing systems in terms of performance. The main reasons for obtaining these achieved results:

Firstly, the design of the Convolutional Neural Network architecture is shown in Figure (3.1), in which identical blocks of layers were stacked to capture discriminative features, and the batch normalization layer was used after each convolutional layer to prevent overfitting and improve accuracy. In addition, we used the early stopping technique that also prevents the overfitting state.

Secondly, the network was trained on segmented images as a region of interest (ROI), and this gives higher accuracy results because it focused on iris tissue only.

Thirdly, the number of parameters in the proposed model reached (4,882,920), which is not much compared to the other literature models.

I. CNN +SoftMax Classifier

By comparing the proposed method (**CNN +SoftMax Classifier**) with previous research in this field, it was noted that this method out performed these researches, and the reason is due to the superiority of the proposed CNN network in training and testing data as shown in table (4.16).

Table (4.16): Comparison of the proposed method (CNN & SoftMax Classifier) with the previous methods

Reference	***CNN Model with SoftMax***	***Dataset***	***Accuracy%***
[7] 2019	VGG-16	CASIA-V1	98
[32] 2021	Mini-VGG Net	CASIA-V1	98
[11] 2021	CNN network	CASIA-V1	95.4
Proposed Method	**Proposed CNN**	**CASIA-V1**	**98.89**
		CASIA-V4	**98.5**

II. Feature fusion & Matching by Euclidean Distance

By comparing this proposed method (**Feature Fusion**) with previous research in this field, it was noted that this method greatly outperformed others, and the reason is due to the superiority of the CNN network in extracting features, as it greatly outperformed the traditional methods as shown in table (4.17).

Table (4.17): Comparison of the Proposed Method (Feature fusion & Matching by Euclidean Distance) with the previous Methods

Methods	*Features Extraction& Similarity Measures*	*Database*	*Accuracy%*
[82] 2015	Radon Transform Thresholding (RTT) & Gradient-based Isolation (GI).& Euclidian Distance	CASIA- Interval	84.17
[19] 2018	Principle Component Analysis (PCA).& Euclidian Distance	CASIA-V4	85.0
[83] 2017	Multi-channel Gaber Filters & Euclidian Distance	CASIA-V1	93.34
[84] 2018	Fourier Descriptors (FD)	CASIA-V1	94
	Principle Component Analysis (PCA) Euclidian Distance		92
Proposed method	**CNN Network& Euclidian Distance**	**CASIA-V1**	**96.29**
		CASIA-V4	**95.83**

III. CNN & Classification by SVM

This method was also compared with a previous search for the same databases and showed its superiority as shown in the table (4.18).

Table (4.18): Comparison of the Proposed Method (CNN & Classification by SVM) with the previous Methods

Reference	***CNN Model with SVM***	***Database***	***Accuracy (%)***
[18] 2018	Pre-trained Alex-Net	CASIA-V1	98.33
		CASIA-Iris Interval	86.6
Proposed System	**Proposed CNN model**	**CASIA-V1**	**99.54**
		CASIA-V4	**98.83**

Chapter Five

Conclusions and Future Work

Chapter Five

Conclusions and Future Works

A. Conclusions

The mean conclusions of the results of using the proposed iris recognition system are as follows:

1- Pre-processing represented by the segmentation and normalization stages is a very important process to determine the iris region of the eye, where the features of this region are extracted only using the Convolutional Neural Network (CNN) without going into details about the features of other parts of the eye. This has increased the accuracy of the iris identification system.

2- Through the results, it was found that the Circular Hough Transform (CHT) along with the image processing techniques have outperformed the other segmentation methods, because the use of image processing techniques has solved many of the problems that the eye's image suffers from, and thus made it easier for Circular Hough Transform to determine the iris accurately.

3- By building a CNN model and training it on the data used, it was found that adjusting the hyper-parameters such as the number of epochs, the number of filters and their size have a role in increasing the accuracy of the CNN.

4- Through the application of the feature fusion technique, it was found that the template that was built by integrating the features extracted from CNN and matching it with the test features give higher accuracy than the built templates from the features extracted from the traditional methods.

5- By experimenting with other classification methods, it was noted that the SVM classifier outperformed other used methods such as SoftMax classifier and feature fusion, due to its high efficiency, which uses part of the decision-making data, the best level depending on the data points on supportive vector and not depending on the entire training evidence.

B. Recommendations for future works

1- The proposed system can be developed to work on larger databases containing thousands of images.

2- Proposed system serves to identify a person (Identification), it can be developed to verify the identity of people (Verification).

3- Other methods of fusion features (such as mean, max, ... etc.) can be implemented in which a template for each person can be built with high accuracy.

4- Other distance measures can be used to determine whether the classification accuracy is greater or less

References

References

[1] I. Buciu and A. Gacsadi, “Biometrics systems and technologies: A survey,” *International Journal of Computers, Communications and Control*, vol. 11, no. 3. pp. 315–330, 2016. doi: 10.15837/ijccc.2016.3.2556.

[2] R. Hentati, M. Hentati, and M. Abid, “Development a New Algorithm for Iris Biometric Recognition,” *International Journal of Computer and Communication Engineering*. pp. 283–286, 2012. doi: 10.7763/ijcce.2012.v1.73.

[3] R. Hidayat and K. Ihsan, “Robust Feature Extraction and Iris Recognition for Biometric Personal Identification,” in *Biometric Systems, Design and Applications*, InTech, 2011. doi: 10.5772/18374.

[4] A. S. Al-Waisy, R. Qahwaji, S. Ipson, S. Al-Fahdawi, and T. A. M. Nagem, “A multi-biometric iris recognition system based on a deep learning approach,” *Pattern Anal. Appl.*, vol. 21, no. 3, pp. 783–802, Aug. 2018, doi: 10.1007/s10044-017-0656-1.

[5] G. Kumar and P. K. Bhatia, “A Detailed Review of Feature Extraction in Image Processing Systems,” in *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, Feb. 2014, pp. 5–12. doi: 10.1109/ACCT.2014.74.

[6] M. G Alaslni and L. A. Elrefaei, “Transfer Learning with Convolutional Neural Networks for IRIS Recognition,” *International Journal of Artificial Intelligence & Applications*, vol. 10, no. 5. pp. 49–66, 2019. doi: 10.5121/ijaia.2019.10505.

[7] M. G Alaslani and L. A. Elrefaei, “Convolutional Neural Network Based Feature Extraction for IRIS Recognition,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 10, no. 2, pp. 65–78, Apr. 2018, doi: 10.5121/ijcsit.2018.10206.

[8] S. Minaee, A. Abdolrashidi, and Y. Wang, “Iris recognition using scattering transform and textural features,” *2015 IEEE Signal Processing and Signal Processing Education Workshop, SP/SPE 2015*. pp. 37–42, 2015. doi: 10.1109/DSP-SPE.2015.7369524.

[9] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.2200/S00822ED1V01Y201712COV015.

[10] Z. Rahim, H. Kadhim, and M. Salih, “Survey of Iris Recognition using Deep Learning Techniques,” *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 13, no. 3. pp. 47–56, 2021.

[11] K. Okokpujie, E. Noma-Osaghae, S. John, and A. Ajulibe, “An improved iris segmentation technique using circular hough transform,” *Lecture Notes in Electrical Engineering*, vol. 450. pp. 203–211, 2017. doi: 10.1007/978-981-10-6454-8_26.

[12] S. S. D. and S. A. Ahmed AK. Tahir, “an-iris-recognition-system-using-a-new-method-of-iris-localization.” International Journal of Open Information Technologies, 2021.

[13] Y. K. Tariq M. Khan, Donald G. Bailey, “A fast and accurate iris segmentation method using an LoG filter and its zero-crossings.” arXiv preprint arXiv:2201.06176., 2022.

[14] J. Lozej, B. Meden, V. Struc, and P. Peer, “End-to-End Iris Segmentation Using U-Net,” in *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, Jul. 2018, pp. 1–6. doi: 10.1109/IWOBI.2018.8464213.

[15] W. Zhang, X. Lu, Y. Gu, Y. Liu, X. Meng, and J. Li, “A Robust Iris Segmentation Scheme Based on Improved U-Net,” *IEEE Access*, vol. 7, pp.

85082–85089, 2019, doi: 10.1109/ACCESS.2019.2924464.

[16] Y. H. Li, P. J. Huang, and Y. Juan, “An Efficient and Robust Iris Segmentation Algorithm Using Deep Learning,” *Mobile Information Systems*, vol. 2019. 2019. doi: 10.1155/2019/4568929.

[17] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, “Iris Recognition With Off-the-Shelf CNN Features: A Deep Learning Perspective,” *IEEE Access*, vol. 6, pp. 18848–18855, 2018, doi: 10.1109/ACCESS.2017.2784352.

[18] M. G Alaslani and L. A. Elrefaei, “Convolutional Neural Network Based Feature Extraction for IRIS Recognition,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 10, no. 2, pp. 65–78, Apr. 2018, doi: 10.5121/ijcsit.2018.10206.

[19] Y. Sari, M. Alkaff, and R. A. Pramunendar, “Iris recognition based on distance similarity and PCA,” *AIP Conf. Proc.*, vol. 1977, no. June, 2018, doi: 10.1063/1.5042900.

[20] T. Zhao, Y. Liu, G. Huo, and X. Zhu, “A Deep Learning Iris Recognition Method Based on Capsule Network Architecture,” *IEEE Access*, vol. 7. pp. 49691–49701, 2019. doi: 10.1109/ACCESS.2019.2911056.

[21] S. Umer, A. Sardar, B. C. Dhara, R. K. Rout, and H. M. Pandey, “Person identification using fusion of iris and periocular deep features,” *Neural Networks*, vol. 122, pp. 407–419, 2020, doi: 10.1016/j.neunet.2019.11.009.

[22] K. Kranthi Kumar, R. Bharadwaj, S. Ch, and S. Sujana, “Effective Deep Learning approach based on VGG-Mini Architecture for Iris Recognition,” *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 5. pp. 4718–4726, 2021.

[23] D. V. R. S.Sujana, “An Effective CNN based Feature Extraction Approach for Iris Recognition System.” Turkish Journal of Computer and Mathematics Education, pp. 4595–4604, 2021.

[24] A. N. Hashim and R. R. Al-Khalidy, “Iris Identification Based on the Fusion

of Multiple Methods," *Iraqi J. Sci.*, pp. 1364–1375, Apr. 2021, doi: 10.24996/ijs.2021.62.4.32.

[25] K. Yang, Z. Xu, and J. Fei, "DualSANet: Dual spatial attention network for iris recognition," *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*. pp. 888–896, 2021. doi: 10.1109/WACV48630.2021.00093.

[26] S. Ismail, F. Hani, M. Ali, and S. A. Aljunid, "The Performance of Deep Neural Networks in Deformed Iris Recognition System," vol. 6, no. 3, pp. 8517–8529, 2022.

[27] I. A. Abdul-mune, "Hand Matching System Based on CNN," *MSc thesis , Univ. Kufa*, 2021.

[28] "Examples-of-biometric-modalities-that-can-be-used-to-authenticate-an-individual." [Online]. Available: https://www.researchgate.net

[29] S. Chokkadi1, S. MS2, S. K. B3, and A. Bhandary4, "A Study on various state of the art of the Art Face Recognition System using Deep Learning Techniques." a Chokkadi et al., International Journal of Advanced Trends in Computer Science and Engineering, 8(4), July- August 2019, 1590, 2019.

[30] M. Roshini1, D. G. Vasanth2, and D. G. N. Kodandaramaiah3, "Recent Segmentation Techniques in Iris Recognition." International Journal of Engineering Research and Applications, 2021. doi: 10.9790/9622-1108010108.

[31] G. Arora, S. Kalra, A. Bhatia, and K. Tiwari, "PalmHashNet: Palmprint Hashing Network for Indexing Large Databases to Boost Identification," *IEEE Access*, vol. 9, pp. 145912–145928, 2021, doi: 10.1109/ACCESS.2021.3123291.

[32] B. Ayotte, M. K. Banavar, D. Hou, and S. Schuckers, "Group leakage overestimates performance: A case study in keystroke dynamics," *IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 1410–1417, 2021. doi: 10.1109/CVPRW53098.2021.00156.

[33] S. C. Yow and A. N. Ali, “Iris Recognition System (IRS) Using Deep Learning Technique,” *Journal of Engineering Science*, vol. 15, no. 2. pp. 125–144, 2019. doi: 10.21315/jes2019.15.2.9.

[34] “Iris-recognition-system-5 (2).” [Online]. Available: https://www.researchgate.net/figure/Iris-recognition-system-5_fig2_349073977%0A

[35] B. Omelina, L., Goga, J., Pavlovicova, J., Oravec, M., & Jansen, “A survey of iris datasets.” Elsevier B.V., 2021. [Online]. Available: http://creativecommons.org/licenses/by/4.0/

[36] A. Nixon, M., & Aguado, “Feature Extraction and Image Processing for Computer Vision.” Acadimic press, 2019.

[37] R. Uhl, A., Busch, C., Marcel, S., & Veldhuis, “Handbook of Vascular Biometrics,” *Image Registration Principles, Tools and methods*. pp. 7–66, 2020. [Online]. Available: http://www.springerlink.com/index/10.1007/978-1-4471-2458-0

[38] I. Petrov and N. Minakova, “Optimization method for non-cooperative iris recognition task using Daugman integro-differential operator,” *J. Phys. Conf. Ser.*, vol. 1615, no. 1, p. 012007, Aug. 2020, doi: 10.1088/1742-6596/1615/1/012007.

[39] “Wikipedia.” [Online]. Available: https://www.wikipedia.org/

[40] K. Adem, “Exudate detection for diabetic retinopathy with circular Hough transformation and convolutional neural networks,” *Expert Syst. Appl.*, vol. 114, pp. 289–295, Dec. 2018, doi: 10.1016/j.eswa.2018.07.053.

[41] “Circular Hough Transform.” [Online]. Available: https://miro.medium.com/max/1400/1*JxcdqbqV8p4zjQDN5qWbiQ.png

[42] N. Cherabit, F. Zohra Chelali, and A. Djeradi, “Circular Hough Transform for Iris localization,” *Sci. Technol.*, vol. 2, no. 5, pp. 114–121, Dec. 2012, doi: 10.5923/j.scit.20120205.02.

[43] A. Asokan, D. E. Popescu, J. Anitha, and D. J. Hemanth, “Bat algorithm based non-linear contrast stretching for satellite image enhancement,” *Geosci.*, vol. 10, no. 2, pp. 1–12, 2020, doi: 10.3390/geosciences10020078.

[44] U. G. Mada, “Contrast Stretching,” *Universitas Stuttgart*. 2009. [Online]. Available: https://uotechnology.edu.iq/ce/lecture 2013n/4th Image Processing _Lectures/DIP_Lecture5.pdf

[45] C. C. Yang, “Image enhancement by modified contrast-stretching manipulation,” *Optics and Laser Technology*, vol. 38, no. 3. pp. 196–201, 2006. doi: 10.1016/j.optlastec.2004.11.009.

[46] “contraststretch histogram.” [Online]. Available: https://staff.fnwi.uva.nl/r.vandenboomgaard/ComputerVision/LectureNotes/IP/PointOperators/ImageStretching.html

[47] J. L. Chen, C. H. Huang, Y. C. Du, and C. H. Lin, “Combining fractional-order edge detection and chaos synchronisation classifier for fingerprint identification,” *IET Image Process.*, vol. 8, no. 6, pp. 354–362, 2014, doi: 10.1049/iet-ipr.2012.0660.

[48] P. Gupta and V. Gaidhane, “A new approach for flame image edges detection,” *International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014*. 2014. doi: 10.1109/ICRAIE.2014.6909178.

[49] M. Kalbasi and H. Nikmehr, “Noise-Robust, Reconfigurable Canny Edge Detection and its Hardware Realization,” *IEEE Access*, vol. 8, pp. 39934–39945, 2020, doi: 10.1109/ACCESS.2020.2976860.

[50] M. P. Deisenroth, R. D. Turner, M. F. Huber, U. D. Hanebeck, and C. E. Rasmussen, “Robust Filtering and Smoothing with Gaussian Processes,”

IEEE Trans. Automat. Contr., vol. 57, no. 7, pp. 1865–1871, Jul. 2012, doi: 10.1109/TAC.2011.2179426.

[51] H. Zhao, R. Yao, L. Xu, Y. Yuan, G. Li, and W. Deng, "Study on a novel fault damage degree identification method using high-order differential mathematical morphology gradient spectrum entropy," *Entropy*, vol. 20, no. 9. 2018. doi: 10.3390/e20090682.

[52] H. hou, Y., Gao, B., Zhang, Q., Yao, P., Geng, Y., Li, X., ... & Wu, "Application of mathematical morphology operation with memristor-based computation-in-memory architecture for detecting manufacturing defects." Fundamental Research, pp. 123–130, 2022.

[53] W. Burger and M. J. Burge, *Principles of Digital Image Processing*. London: Springer London, 2013. doi: 10.1007/978-1-84882-919-0.

[54] L. HE, Y. CHAO, X. ZHAO, B. YAO, H. KASUYA, and A. OHTA, "An Algorithm for Calculating Objects' Shape Features in Binary Images," *DEStech Trans. Comput. Sci. Eng.*, no. aiea, pp. 1053–1057, 2017, doi: 10.12783/dtcse/aiea2017/15047.

[55] H. Zhang, D. Wang, and K. Song, "An improved fast algorithm for searching centroid of object in binary image," *Third Int. Symp. Multispectral Image Process. Pattern Recognit.*, vol. 5286, no. September 2003, p. 859, 2003, doi: 10.1117/12.538711.

[56] A. L. and L. Nanni, "'Overview of the combination of biometric matchers,' Information Fusion." pp. 71–85, 2017.

[57] H. I. Abdulrazzaq, "Robust Authorization Based on Multi Biometric Techniques," 2018.

[58] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in Deep Learning*, vol. 57, no. January. 2019. doi: 10.1007/978-981-13-6794-6.

[59] X. Serra and J. Castán, "Face recognition using Deep Learning," *Catalonia:*

Polytechnic University of Catalonia, vol. 78, no. January. 2017.

[60] F. Del Frate, F. Pacifici, G. Schiavon, and C. Solimini, “Use of neural networks for automatic classification from high-resolution images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4. pp. 800–809, 2007. doi: 10.1109/TGRS.2007.892009.

[61] S. K. Kaliyugarasan, “Deep transfer learning in medical imaging-MS thesis. The University of Bergen,” no. June. 2019.

[62] B. Karlik and A. V. Olgac, “Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks,” *And Expert Systems (IJAE)*. pp. 111–122, 2011.

[63] S. Ruder, “An Overview Optimization Gradients,” *arXiv preprint arXiv:1609.04747*. pp. 1–14, 2017. [Online]. Available: https://arxiv.org/pdf/1609.04747.pdf

[64] “Deep-Neural-Network-architecture.” https://www.researchgate.net/publication/330120030/figure/fig1/AS:735637925797888@1552401157053/Deep-Neural-Network-architecture.ppm.

[65] H.-C. Shin *et al.*, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning,” *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016, doi: 10.1109/TMI.2016.2528162.

[66] “Full-CNN-overview.” [Online]. Available: https://cdn.analyticsvidhya.com/wp-content/uploads/2021/05/Full-CNN-overview.png

[67] Hung Dao, “IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS.” Information Technology Oulu University of Applied Sciences, 2020.

[68] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, “Fault Diagnosis for

Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 101–110, Feb. 2018, doi: 10.1109/TMECH.2017.2728371.

[69] J. Heidari, "Classifying Material Defects with Convolutional Neural Networks and Image Processing, MSC Thesis." UPPSALA University, 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-387797%0ARefrence66%0A

[70] Gong *et al.*, "A Novel Deep Learning Method for Intelligent Fault Diagnosis of Rotating Machinery Based on Improved CNN-SVM and Multichannel Data Fusion," *Sensors*, vol. 19, no. 7, p. 1693, Apr. 2019, doi: 10.3390/s19071693.

[71] "maxpooling." [Online]. Available: https://media.geeksforgeeks.org/wp-content/uploads/20190721025744/Screenshot-2019-07-21-at-2.57.13-AM.png

[72] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*. 2015.

[73] "flatten plot." [Online]. Available: https://www.alrab7on.com/artificial-neural-networks/

[74] G. Allen, R. De Veaux, and R. Nugent, "Springer Texts in Statistics Series Editors." 2021. [Online]. Available: http://www.springer.com/series/417

[75] "overfitting." [Online]. Available: https://miro.medium.com/max/567/1*2BvEinjHM4SXt2ge0MOi4w.png

[76] R. J. T. Alnur Ali, J. Zico Kolter, "A Continuous-Time View of Early Stopping for Least Squares." Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, PMLR 89:1370-1378, 2019.

[77] V. Varkarakis, S. Bazrafkan, and P. Corcoran, "Deep neural network and data augmentation methodology for off-axis iris segmentation in wearable headsets," *Neural Networks*, vol. 121. pp. 101–121, 2020. doi: 10.1016/j.neunet.2019.07.020.

[78] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15. pp. 1929–1958, 2014.

[79] S. Thiyagarajan, "Performance Comparison of Hybrid CNN-SVM and CNN-XGBoost models in Concrete Crack Detection." 2019. [Online]. Available: https://arrow.tudublin.ie/scschcomdis

[80] M. Edida, N. J. Lakshmi, and N. Jabalia, "Automated Diagnosis of Diabetes Mellitus Based on Machine Learning," *Artif. Intell. Mach. Learn. Healthc.*, pp. 37–56, 2021, doi: 10.1007/978-981-16-0811-7_2.

[81] "CASIA Iris Image Databases." http://biometrics.idealtest.org.

[82] B. V. Bharath, A. S. Vilas, K. Manikantan, and S. Ramachandran, "Iris recognition using radon transform thresholding based feature extraction with gradient-based isolation as a pre-processing technique," *9th International Conference on Industrial and Information Systems, ICIIS 2014*. 2015. doi: 10.1109/ICIINFS.2014.7036572.

[83] P. D. Mane, K. S. Bawankar, S. S. Jagtap, and U. R. Kamble, "Multichannel Gabor based Iris Recognition," no. February, 2017.

[84] M. H. Hamd and S. K. Ahmed, "Biometric System Design for Iris Recognition Using Intelligent Algorithms," *Int. J. Mod. Educ. Comput. Sci.*, vol. 10, no. 3, pp. 9–16, 2018, doi: 10.5815/ijmecs.2018.03.02.

المستخلص

يوفر نظام القياسات الحيوية الية تحديد هوية الفرد بناءً على ميزة أو خاصية فريدة يمتلكها الفرد. يعد التعرف على قزحية العين أحد أكثر تقنيات تحديد الهوية تمثيلا في التعرف على القياسات الحيوية ، والذي يستخدم على نطاق واسع في مختلف المجالات. تعتمد هذه الأنظمة على سمات نسيج قزحية العين ، والتي تتميز بالثراء والعشوائية والتفرد ، بالإضافة إلى الثبات ، وهذا يجعل أنظمة التعرف على قزحية العين أكثر دقة وموثوقية.

المشاكل في صور قزحية العين (مثل الجودة الرديئة وصور قزحية غير خطية مشوهة) تجعل مهمة التعرف أكثر صعوبة وتحديًا. كما أن الضوضاء المتأصلة في صور قزحية العين تسبب تدهورًا كبيرًا في كفاءة التمييزاو التعرف على الشخص. كل هذه المشاكل تفتح تحديات في موضوع التعرف على قزحية العين. وبالتالي ، من المهم تطوير نظام فعال ودقيق للتعرف على قزحية العين يعزز دقة التعرف. لتلبية هذه الاحتياجات ، في هذه الرسالة ، يتم اقتراح نظام تحديد قزحية العين الذي يعتمد على العثور على مجموعة من ميزات قزحية العين التي تعزز دقة النظام.

في السنوات الأخيرة ، حقق التعلم العميق أداءً عالياً في العديد من مهام الرؤية الحاسوبية ، مثل تصنيف الصور ، والتجزئة الدلالية ، واكتشاف الأشياء. لذلك ، تم الاعتماد على تقنيات التعلم العميق لاستخراج ملامح نسيج القزحية. لاستخراج الميزات ، يتم استخدام تقنية التعلم العميق التي تمثلها الشبكة العصبية التلافيفية (CNN). تم تصميم شبكة CNN بحيث تتكون من عدة طبقات لاستخراج الميزات.

من خلال مراحل التدريب والاختبار ، يتم تمرير صورة العين المدخله عبر سلسلة من مراحل المعالجة الرئيسية (وهي تحديد القزحية, تحويل الشكل الدائري للقزحية إلى مستطيل واستخراج الخصائص). ان تحديد القزحية هو عملية فصل منطقة القزحية عن باقي أجزاء العين. يتم تحديد منطقة قزحية العين من خلال إيجاد حدود البؤبؤ أولاً باستخدام مجموعة من عمليات معالجة الصور. ثانيًا ، يتم تطبيق طريقة لتحديد الحدود الخارجية لمنطقة القزحية. لتجاوز المشاكل التي تنشأ بسبب التغيير الطبيعي في حجم البؤبؤ ، يتم إعادة تشكيل القزحية الدائرية وتعيينها لتكون مستطيلة. تقع القزحية بين الحدقة والصلبة البيضاء ، لذا فهي ذات شكل حلقي وحجم غير مستقر ، لذلك هناك حاجة لعملية أخرى لتحويلها إلى مستطيل ثابت الأبعاد يسهل استخلاص معالمه. هذه العملية تسمى Normalization.

لإظهار تأثير الميزات المستخرجة بواسطة نموذج CNN المقترح ، تم استخدام عدة طرق لتحديد هوية الشخص. كانت الطريقة الأولى في التصنيف هي استخدام الشبكة العصبية الاصطناعية (ANN) ممثلة بالطبقة المتصلة بالكامل في نموذج CNN ومصنف SoftMax. الثاني يشمل التصنيف باستخدام الحد الأدنى من المسافة الإقليدية. في هذه الطريقة ، يتم استخدام الميزات المستخرجة في مرحلة التدريب لبناء قالب لكل شخص بناءً على تطبيق طريقة اندماج الميزات المعتمدة على حساب متوسط قيم عينات التدريب لكل شخص ويتم حفظها في قاعدة بيانات التدريب. طريقة التصنيف الثالثة هي استخدام آلة متجه الدعم (SVM). تم اختبار النظام المطور على قاعدتي بيانات (CASIA-1 و CASIA-4). كانت نتائج الطريقة الأولى 98.89 و 98.5 ل CASIA-V1 و CASIA-V4 على التوالي. كانت نتائج الطريقة الثانية 96.29٪ و 95.83 لـ CASIA-V1 و CASIA-V4 على التوالي. كانت نتائج الطريقة الثالثة 99.54٪ و 98.83٪ للطريقة CASIA-V1 و CASIA-V4 على التوالي.

جمهورية العراق

وزارة التعليم العالي و البحث العلمي

جامعة بابل – كلية العلوم للبنات

قسم علوم الحاسوب

تحديد هوية الشخص بناءً على الاندماج العميق لميزات القزحية وأساليب التعلم الآلي

رسالة مقدمة الى

مجلس كلية العلوم للبنات-جامعة بابل

وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

اسراء عادل حسن

بإشراف

د.سهاد احمد علي

د.هضاب خالد عبيس

1444 2022