

**ANKARA YILDIRIM BEYAZIT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**



**IRIS IDENTIFICATION BASED ON DEEP
CONVOLUTIONAL NEURAL NETWORKS**

M.Sc. Thesis by

Ahmad SALIHY

Department of Computer Engineering

September, 2023

ANKARA

IRIS IDENTIFICATION BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS

**A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of
Ankara Yildirim Beyazit University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in
Computer Engineering, Department of Computer Engineering**

**by
Ahmad SALIHY**

**September, 2023
ANKARA**

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**IRIS IDENTIFICATION BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS**" completed by **AHMAD SALIH** under supervision of **PROF. DR. REMZI YILDIRIM** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. REMZI YILDIRIM

Supervisor

Prof.Dr.Bünyamin CEYLAN

Jury Member

Dr.Öğr.Üyesi Mustafa YENİAD

Jury Member

Prof. Dr. Sadettin ORHAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 2023, 20 September

Signature: _____

Name & Surname: Ahmad SALIHY

ACKNOWLEDGEMENTS

This Thesis, is the dissertation of my postgraduate degree and the study I have selected is the; Iris identification based on Deep Convolutional Neural Network, which tends toward itself during the postgraduate periods. This interest has inspired me to proceed especially for this study to the end of my postgraduate education period.

I would like to express my profound gratitude and immense thanks to my supervisor Professor Dr. Remzi Yıldırım. Delay for all his concerned advice, guidance, support, and encouragement throughout the period of this work. I have been highly privileged to know him and work under his supervision and mentorship. I am also profoundly grateful to Professor Dr. Fatih Çelibi for his guidance, excellent discussions, and positive criticisms of my work which have helped and supported me throughout the period of this research.

At last, it's necessary to be grateful to those, who supported and helped me during these three academic years to achieve success in my studies and research, and I am really thankful to my respected instructors, who honestly tried to share their scientific hoards without any referring of our positions, respected classmates those had impressed on amelioration of behavior, sermon and acting, respected friends those who encourage, instructed and provided essential resources during these three years, Respected Professor Dr. Remzi Yildir, whom he helped me with his overall useful guides and all those, whom they directly or indirectly helped me and wish them happy moments of life.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved family for their blessings, and our friends/classmates for their help and wishes for the successful completion of this study.

September, 2023

Ahmad SALIHY

IRIS IDENTIFICATION BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS

ABSTRACT

With the prominent need for security and the increasing request for information security, security regulations, and trustworthy methods of identification and adjustment in biometric systems worldwide, biometric identification technology transpires extensively used in our everyday lives. With the advancement of science and technology iris biometrics analogous to many further biometric systems offers an alternative solution to this lingering matter. Although iris identification has been immensely studied, it is nevertheless, not a fully solved problem, which is the factor inhibiting its implementation in real-world situations today. The existing iris biometrics technology and other biometrics systems' main problems faced a lack of robustness of the algorithms. But lately, iris identification for biometrics become vigorous by using deep learning. On account of this consideration, iris biometrics technology has gained more attention and interest due to its ability and reliability to dominate a sum of significant limitations of unimodal biometric systems. In this thesis, a new biometric identification is proposed, which is based on a deep learning algorithm (CNN) for identifying humans using biometric modalities of the iris.

In the interpretation of the countless performances of deep learning methods in several identification tasks, the structure of the proposed biometric identification is based on convolutional neural networks (CNNs) which segment, extract features, and classify images using SoftMax and ReLu activation function classifiers. Furthermore. VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet, and InceptionV3 were performed for feature extraction, which attained around 97.68% accuracy on the training data set and almost 98.35% accuracy on the test data set. In iris classification, I have used CNN, ImageNet, and TinyVGG which obtained 98.11% accuracy on the training data set and 98.35% on the test data set. As a highly accurate modality of biometric identification, iris identification uses plenty of mathematical pattern recognition techniques on images of one or both of the irises of a person's eyes.

Keywords: identification, biometric identification system, security, technology, deep learning, convolutional neural networks, computer vision, iris identification system.

DERİN KONVOLÜSYONEL SİNİR AĞLARINA DAYALI İRİS TANIMLAMASI

ÖZ

Dünyada, güvenlik ihtiyacı değişik yöntemler kullanılarak yapılmaktadır. Biyometrik sistemleri de çok değişik amaç için kullanılmaktadır. Bilim ve teknolojinin gelişmesiyle birlikte, bir çok biyometrik sisteme alternatif olarak iris biyometrisi bir çözüm sunuyor. Bu araştırmada, irisin biyometrik modalitelerini kullanarak insanları tanımlamak için derin öğrenme algoritmasına (CNN) dayanan yeni bir biyometrik tanımlama önerilmiştir. Çalışma görevlerinde derin öğrenme yöntemlerinin sayısız performansının yorumlanması, önerilen biyometrik tanımlamanın yapısı, SoftMax ve ReLu aktivasyon fonksiyon sınıflandırıcılarını kullanarak görüntüleri bölümlere ayıran, öznitelikleri çıkaran ve sınıflandıran evrişimli sinir ağları (CNN'ler) ve VGG-16, VGG-19, ResNet-50 ve ResNet101 kullanılmıştır. Çalışma sonucunda, özellik çıkarımı için MobileNet, AlexNet ve InceptionV3 gerçekleştirildi ve eğitim verilerinde %97,68 ve test verilerinde ise %98,35 doğruluk elde edilmiştir. İris sınıflandırmasında ise CNN, ImageNet ve TinyVGG kullanılarak eğitim test verilerin de %98.11 ve test verilerinde %98.35 doğruluk elde edilmiştir.

Anahtar kelimeler: tanımlama, biyometrik tanımlama sistemi, güvenlik, teknoloji, derin öğrenme, evrişimli sinir ağları, bilgisayar görüşü, iris tanımlama sistemi.

CONTENTS

M.Sc. THESIS EXAMINATION RESULT FORM	ii
ETHICAL DECLARATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
ÖZ	vi
NOMENCLATURE.....	xi
LIST OF TABLES.....	xiii
LIST OF FIGURES	xvii

CHAPTER 1 – INTRODUCTION.....1

1.1 Biometric.....	1
1.2 Motivation and Challenges.....	4
1.2.1 Lack of Robustness.....	6
1.2.2 Non-Circularity of the Iris Boundaries	6
1.2.3 Speed	6
1.2.4 Dimensionality	7
1.3 Iris Biometrics Overview	8
1.3.1 Features of the Iris.....	9
1.4 The proposed Iris Identification	11
1.5 Strengths and Weaknesses of the Iris Identification Biometric	12
1.5.1 Strengths of Iris Biometrics.....	12
1.5.2 Weaknesses of Iris Biometrics	13
1.6 Objectives and Aims of the Research	13
1.7 Structure of the Thesis	15

CHAPTER 2 – Background Study and Literature Review 16

2.1 Deep Convolutional Neural Network Background	16
2.2 Iris as Biometric Traits	18
2.3 Iris Image Preprocessing	18
2.3.1 Gaussian Filter.....	19

2.3.2 Hough Transforms	20
2.4 Iris Image Segmentation	21
2.4.1 Daugman's Differential Operator.....	22
2.4.2 Wildes' Hough Transform.....	22
2.4.3 U-Net Iris Image Segmentation.....	23
2.5 Iris Image Normalization	23
2.5.1 Homogeneous Rubber Sheet Model	24
2.6 Iris Feature Extraction.....	25
2.6.1 Histogram of Oriented Gradient (HOG)	26
2.6.2 VGG – 16 and VGG – 19 Models	26
2.6.3 ResNet Models	28
2.7 Iris Pattern Matching	29
2.7.1 ORB Algorithm	29
2.8 Different Biometric Technologies.....	30
2.8.1 Face Recognition	32
2.8.2 Voice Recognition	34
2.8.3 Hand Geometry and Finger Print.....	37
2.9 Related Works	40
2.9.1 History of Iris Biometrics	40
2.9.2 Review of Selected Relevant Literatures	44
CHAPTER 3 – Methodology	53
3.1 Deep Learning and Convolutional Neural Network in Iris Identification.....	53
3.1.1 Convolution Layers.....	55
3.1.2 Pooling Layers	56
3.1.3 Fully Connected Layers	56
3.2 Iris Identification Phases	57
3.2.1 Iris Image Acquisition	57
3.2.1.1 Iris Databases	59
3.2.2 Iris Image Preprocessing.....	59
3.2.2.1 Iris Image Filtering	61

3.2.2.2 Edge Detection.....	63
3.2.3 Iris Image Segmentation	66
3.2.4 Iris Image Normalization	70
3.2.5 Iris Image Feature Extraction	73
3.2.6 Iris Image Pattern Matching	77
3.2.7 Iris Classification	79
CHAPTER 4 – Experimental Results and Discussion	83
4.1 Iris Databases	83
4.1.1 Uiris Version 2 (v2)	83
4.1.2 Multimedia University (MMU)	84
4.1.3 Gaze Interaction for Everybody (GI4E)	84
4.2 Implementation and Experimental Results	84
4.2.1 Evaluation of the Iris Preprocessing Scheme	85
4.2.2 Accuracy and Evaluation of the Iris Segmentation Scheme	86
4.2.3 Feature Extraction Evaluation Scheme	90
4.2.4 Evaluation of the Iris Classification scheme	94
CHAPTER 5 – Conclusion	99
REFERENCES.....	103
APPENDICES.....	122
Appendix A – First Appendix	123
Appendix B – Second Appendix	125
Image preprocessing Software Code	125
Image Segmentation Software Code	134
Image Feature Extraction Software Code	137
Image Classification Software Code.....	139
Appendix C – Third Appendix.....	142
Theoretical Background of Iris Biometric Methods	143

Mathematical Methods	144
CURRICULUM VITAE	153



NOMENCLATURE

Acronyms

ACC	Accuracy
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
API	Application Programming Interface
ATMs	Automated Teller Machines
BATH	University of Bath
BN	Batch Normalization
CASIA	Chinese Academy of Sciences' Institute of Automation
CCD	charge-coupled device
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
DA	Disadvantage
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DWT	Discrete Wavelet Transforms
GI4E	Gaze Interaction for Everybody
GUI	Graphical User Interface
HID	High Intensity Discharge
HOG	Histogram of Oriented Gradient
ML	Machine Learning
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MTS	Model Training Sets
NIR	Near InfraRed Spectroscopy
NumPy	Numerical Python
ORB	Oriented FAST and rotated BRIEF
PERF	Performance

ReLU	Rectified Linear Units
ResNet	Residual Network
RGB	Red Green Blue
SNR	Signal-To-Noise Ratio
ST	Standard Deviation
STD	Standard Deviation
SVM	Support Vector Machines
TL	Transfer Learning
UPOL	University of Olomouc
USD	United States Dollar
VGG	Visual Geometry Group

LIST OF TABLES

Table 2.1	Layer Description of CNNs [93]	40
Table 2.2	Implication of Error Rates[74]	40
Table 2.3	Training Parameters of CNNs [93]	41
Table 2.4	Contactless Biometrics Market Report Scope [77]	44
Table 2.5	Comparison table on literature survey[74]	51
Table 2.6	Comparison between pertained models[66]	51
Table 2.7	Iris Segmentation Accuracy Result on different Methods[51]	52
Table 3.1	Segmenation Model Training Result (%).....	69
Table 3.2	Segmenation Model Training Result (%).....	70
Table 4.1	Iris Segmentation Model Training Comparisons and the Results (%) ...	89
Table 4.2	Feature Extraction Model Training Comparisons and the Results(%) ...	92
Table 4.3	Image Classification Comparisons and the Results (%).....	96
Table C.1	Seven Pillars Of Biometrics System [159]	143

LIST OF FIGURES

Figure 1.1	Iris Image Variability in CASIA and UBIRIS Databases	4
Figure 1.2	The human cortical frontal eye [10]	4
Figure 1.3	The anatomy of the human eye [21]	10
Figure 1.4	The anatomy of the human eye [21]	10
Figure 1.5	Iris Identification stages.....	12
Figure 2.1	Eye Detection image from GI4E database.....	19
Figure 2.2	U-Net Structure used for Iris Image Segmentation.....	24
Figure 2.3	Rubber sheet model [29].....	25
Figure 2.4	HOG Iris Feature Extraction Result	26
Figure 2.5	VGG-16 Model Architecture	27
Figure 2.6	VGG-19 Iris Feature Extraction Model Architecture.....	28
Figure 2.7	Distribution of pre-trained CNN Models	29
Figure 2.8	Distribution of pre-trained CNN Models [69].....	30
Figure 2.9	Bidirectional RNN[87].....	36
Figure 2.10	Features of Fingerprints that are used for classification [90].....	38
Figure 2.11	The graphical illustration of the CNN architecture [92].....	40
Figure 2.12	Contactless Biometrics Market Size [77]	43
Figure 3.1	CNN architecture for Iris identification	54
Figure 3.2	DL and CNN block diagram of a perceptron binary classifier.....	55
Figure 3.3	Convolutional Neural Network Architecture.....	56
Figure 3.4	The proposed Iris Identification Flowchart and Architecture	58
Figure 3.5	The results of the preprocessing stage performed	60
Figure 3.6	Eye and the Iris inner Detection	60
Figure 3.7	Histogram before and after image Exponential Operator	62
Figure 3.8	Histogram Equalization Implementation.....	63
Figure 3.9	Gaussian filtering	64
Figure 3.10	Ellipse Fitting Iris Edge Detection.....	65
Figure 3.11	Iris edge detection by Canny edge detection	65
Figure 3.12	Hough Transform iris outer and inner boundary detection	66
Figure 3.13	Block diagram of segmentation stage [21]	67

Figure 3.14	U-Net Flowchart and Structure used for Iris Image Segmentation	68
Figure 3.15	The result of Iris Segmentation using U-Net	69
Figure 3.16	Rubber sheet model [29].....	71
Figure 3.17	The Normalized iris image.....	71
Figure 3.18	HOG Iris Feature Extraction	73
Figure 3.19	HOG Iris Feature Extraction Texture Result	74
Figure 3.20	VGG-16 Iris Feature Extraction Architecture	75
Figure 3.21	VGG-19 Iris Feature Extraction Architecture	76
Figure 3.22	Distribution of pre-trained CNN Models	76
Figure 3.23	Image matching flow chart based on ORB algorithm [69].....	77
Figure 3.24	Matched Keys points of the same iris	78
Figure 3.25	ORB Matched Points on the same iris	79
Figure 3.26	Matched Keys points of different irises.....	79
Figure 3.27	ORB Matched Points on different irises.....	80
Figure 3.28	The accuracy and loss curves of model's training performance.....	81
Figure 3.29	The accuracy and loss curves of model's training performance.....	82
Figure 4.1	Histogram Equalization before and after image filtering	86
Figure 4.2	Hough Transform iris outer and inner boundary detection	86
Figure 4.3	Iris Segmentation result on U-Net model	87
Figure 4.4	Iris Segmentation accuracy and loss curve with U-Net model	88
Figure 4.5	Iris Segmentation accuracy and loss graph with CNN model	88
Figure 4.6	Iris Segmentation accuracy Comparison graph	89
Figure 4.7	Iris Segmentation Error Rate Comparison graph	90
Figure 4.8	Iris Segmentation Standard Deviation Comparison graph	90
Figure 4.9	Model training accuracy and loss result on ResNet50 V2	91
Figure 4.10	Model training accuracy and loss result on EfficientNetB0	92
Figure 4.11	Iris Image Feature Extraction Accuracy Comparison graph.....	93
Figure 4.12	Iris Image Feature Extraction Error Rate Comparison graph	93
Figure 4.13	Iris Image Feature Extraction ST Comparison graph	94
Figure 4.14	The accuracy and loss curves of model's training performance.....	94
Figure 4.15	ImageNet accuracy and loss performance	95
Figure 4.16	TinyVGG accuracy and loss performance	95

Figure 4.17	Iris Image Classification accuracy Comparison graph	96
Figure 4.18	Iris Image Classification Error Rate Comparison graph.....	97
Figure 4.19	Iris Image Classification Standard Deviation Comparison graph	97
Figure A.1	DL, ML, and AI concept [IBM Developer]	123
Figure B.1	Libraries used in image preprocessing.....	126
Figure B.2	Importing the Google Drive	126
Figure B.3	Extracting the compressed Dataset code	126
Figure B.4	Detecting the eye and iris center code	127
Figure B.5	Cropping the eye image from the full-face image	127
Figure B.6	Normalizing the cropped eye images code	128
Figure B.7	Applying Dilation and Gaussian filtering code.....	128
Figure B.8	splitting data into train and test set code	129
Figure B.9	Image data augmentation code.....	129
Figure B.10	Autoencoder Class and the forward method code	130
Figure B.11	CNN model to detect the iris center code.....	131
Figure B.12	Visualizing the predicted iris centers using matplotlib	131
Figure B.13	Code which is Extracting and plotting the histogram	132
Figure B.14	Gaussian filter code which blurs the required region of an image	133
Figure B.15	Hough Transform detecting the outer and inner boundaries of iris....	133
Figure B.16	Ellipse Fitting Iris Edge Detection code	134
Figure B.17	The U-Net architecture code with CNN.....	134
Figure B.18	Compiling the U-Net model	135
Figure B.19	Post Process & Callback class in iris image segmentation.....	135
Figure B.20	U-Net Model Training code with 150 epochs and a batch size of 32 .	136
Figure B.21	Code to visualize the acc and loss for each epoch in segmentation ...	136
Figure B.22	Code to visualize the result of U-Net using matplotlib	137
Figure B.23	Creating data loaders in TL for iris image feature extraction	137
Figure B.24	Creating the Model in TL for iris image feature extraction	138
Figure B.25	Creating the model with ResNet50V2 and compiling the model	138
Figure B.26	Fitting the ResNet50V2 model with 100 epochs	139
Figure B.27	Creating, Compiling and Fitting the CNN image classification	139
Figure B.28	Creating train and test data generators for classification.....	140

Figure B.29 Creating the Model with a 3-layer CNN on ImageNet	140
Figure B.30 Compilling and fitting the model created on ImageNet.....	141
Figure B.31 Creating, Compiling and fitting the model on TinyVGG.....	141



CHAPTER 1

INTRODUCTION

In this chapter, I will provide a brief introduction to iris identification using Deep Convolutional Neural Networks and explain the motivation behind this research. I will also discuss the goals and objectives of this research, along with its contributions. Additionally, this chapter will cover the proposed iris identification structure, strengths, weaknesses, and evaluation methodology.

1.1 Biometric

With the imminent need for security and due to the growing number of requests for data safety, regulations on security, and trustworthy methods of identification and adjustment in the biometric system around the world, biometric identification methods are extensively used in our regular lives. In this research, a new biometric identification is proposed, which is based on a deep learning algorithm for identifying humans using biometric modalities of the iris. Lately, deep learning has had a considerable influence and generated excellent results in biometrics systems [1]. Deep learning algorithms have dominated a large number of the limitations of other machine learning algorithms, principally those related to feature extraction techniques, Deep learning algorithms can cope with biometric image transformations and extract features from raw data [2].

Numerous biometric researchers have trusted machine learning algorithms for identification targets. Machine learning algorithms need extraction techniques to extract features from raw biometric data and transform the raw data into a suitable and reliable format before classifying it [3]. Furthermore, machine learning algorithms necessitate and involve certain preprocessing stages to be derived before the feature extraction. Additionally, some extraction techniques do not continuously work positively with different types of biometrics or different data sets of the same biometrics, moreover, they cannot handle biometric image transformations, for instance, zooming and rotation [4]. In the interpretation of the countless

implementation of CNN, and deep learning algorithms in several identification obligations, the structure and establishment of the proposed iris identification are based on deep convolutional neural networks (CNNs) which has the ability to extract features from images, and classify those images by Sigmoid, SoftMax, and ReLU activation function classifier. However, current advancements and improvements in computer vision and deep learning illustrate that using convolutional neural networks (CNNs) are able to represent complex image characteristics, for feature extraction and classification respectively to increase the efficiency of recognition [3]. Nowadays iris identification system has turned into a reputable and reliable technique for personal and individual identification technique. The system has been used for years in numerous commercial, trade, mercantile, and government applications that allows availability and serviceability inspection and control in localities and places such as automated teller machines (ATMs), office, laboratory, and border control in the airport [4].

Iris identification or iris scanning is an automated method of individual identification or the operation of applying visible and near-infrared light to take a high-contrast photograph of a person's iris which uses mathematical pattern recognition techniques and appears as one of the most effective and beneficial methods for recognition and identification in recent decades. The goal of the iris identification system can be to identify and recognize the human identity and physiological or behavioral characteristics via the textural characteristics of one's iris muscular patterns. As in the 21st century, most people use electronic devices such as smartphones, laptop computers, desktop computers, smartwatches, and tablets, and the usage of these personal devices is growing up. To browse over web-based social platforms and ideology, store personal information such as pictures and videos, and chat with others thru text or video. The quantity of personal data and information stored in electronic devices is increasing day by day. Therefore, biometric authentication is obligatory to avoid unauthorized users and operators from abducting such information and data from personal electronic devices.

The iris is an internal organ of the body that can be captured noninvasively using a

cheap CCD camera [5]. The iris is declared to be one of the most consistent biometric behaviors in existence in today's technology and it is extremely unique to the individual that even the iris patterns of both irises of the same individual and that of identical twins are completely different [6]. However, the particular fascinating features of the human iris have persuaded researchers toward the benefit and utility of the iris patterns for individual authentication and verification. Though, the iris identification and recognition systems involve intricate and intricate mathematical operations, that occasionally could be far slack and slow to be performed in recent extremely challenging and improved systems. Correspondingly, the capture procedure of most applied iris identification systems comprises a stop-and-stare process or head positioning process which is not quite convenient and it might be a waste of time [5]. Furthermore, the environmental circumstance for the detention of iris images involves the use of near-infrared lightening to elucidate the iris before capture which is also a source of error and demands the subject's full cooperation, these are limiting factors for the iris biometric technology [7]. At present, researchers are working on expanding the capabilities of the iris identification technology. It is increasingly common to capture iris images while on the move and from a distance using new techniques. Unfortunately, these methods often result in non-ideal images that are difficult to process with standard iris identification algorithms. Figure 1.1 provides examples of non-ideal iris images captured under various challenging conditions, such as camera noise, reflections, and occlusions.

The Human cortical frontal eye consists of the iris, sclera, and pupil, Figure 1.2 better demonstrates the human frontal eye, the black denotes the pupil, the white illustrates the sclera, and the edge between the sclera and pupil is called the iris [8]. Human irises have an exceptional structure that provides interlacing minute characteristics, such as freckles, coronas, and stripes, these visible characteristics of the iris are generally called iris texture which makes it unique and suitable for biometric measurement [8]. Compared with other biometrics technologies, such as palm print, face recognition, finger geometry recognition, Hand Geometry Recognition, the individuality of the iris pattern makes it further reliable and dependable, and stable for identification [8].

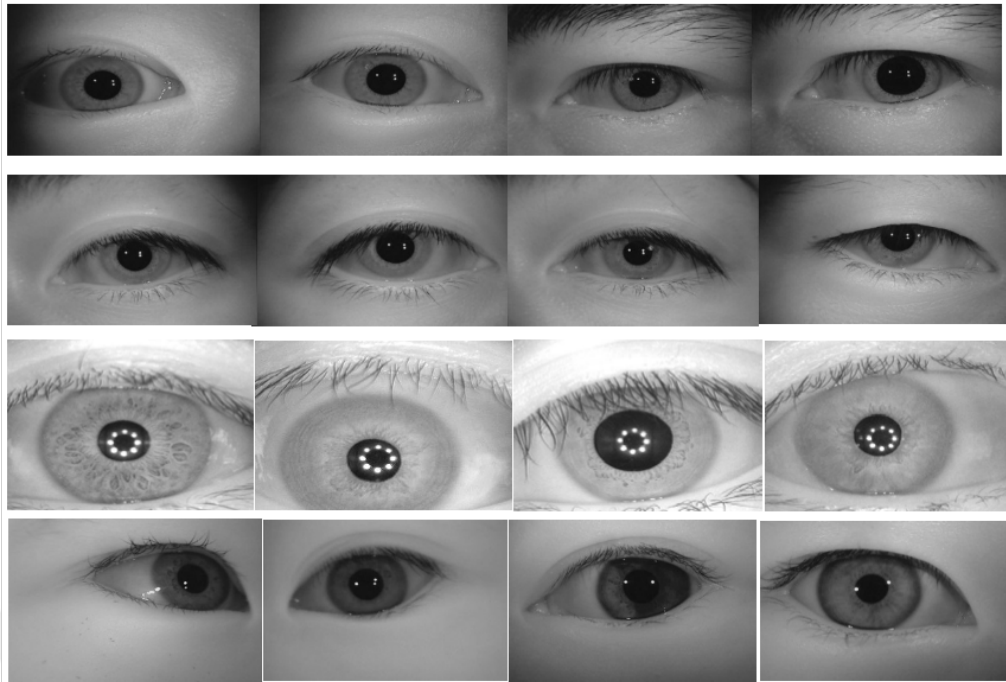


Figure 1.1 Samples of Iris Images images from CASIA AND UBIRIS database [7]

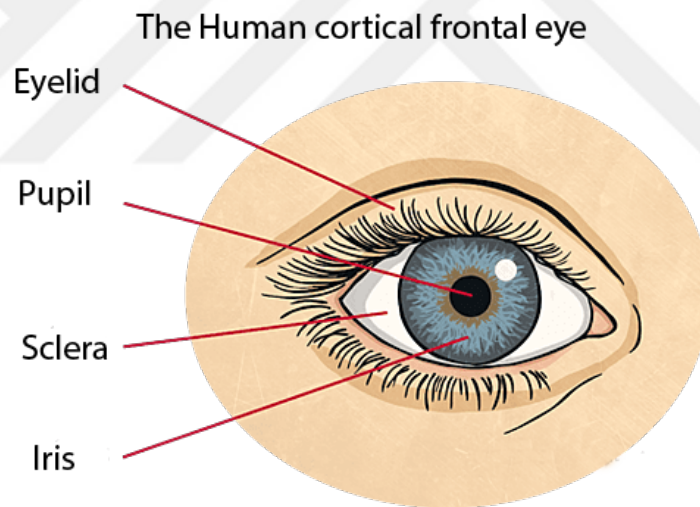


Figure 1.2 The human cortical frontal eye [10]

1.2 Motivation and Challenges

The extraordinary individuality of iris patterns and the ability to capture iris images non-invasively have enthused researchers in iris biometrics, pattern identification, and computer vision to develop an automated system for iris identification based on 2D iris images [9]. While existing iris identification systems have verified and confirmed good performances, the problem of reaching a high identification accuracy in non-ideal

iris images at an effective speed still remains a challenge with no efficient solution up to now [10] [11]. The efficiency and competence of iris identification systems are highly dependent on the accuracy and speed of the iris segmentation module as well as iris feature extraction [12]. Moreover, improving the speed and accuracy of the segmentation modules and feature extraction modules would considerably strengthen and elevate the efficaciousness and efficacy of the iris identification biometric. In typical non-ideal iris image segmentation, the iris boundaries are either segmented through a costly curve evolution technique based on a level-set method [13], or using a supervised learning method [14]. The mentioned methods are statistically costly and persuade deleterious to the complete acceleration of the identification method.

Correspondingly, the extraction of important features from the non-ideal iris pattern is still a big problem that desires a solution, and existing methods of iris feature extraction represent the iris pattern with high dimensional iris code, which negatively affects the classification efficiency [12]. Furthermore, iris images captured with minor quality under lesser or no subject's cooperation is the instruction current moment, the current iris feature extraction algorithms typically lead to numerous untruthful refusal fallacies. Therefore, illustrating the iris identification system with miniature codes is a matter which has been reviewed with no effective resolution so far. Most of the current methods for iris feature extraction and representation have tried to use discrete Wavelets Transform, Gabor Filter, and Laplacian of Gaussian filter to extract iris features [15] [16]. This approach is normally affected by shift variance (if $g(x + x_0) = H[f(x + x_0)]$) and phase problems which are inherent in discrete wavelet transforms (DWT) [8]. Furthermore, a robust identification method needs to be instituted to conquer the competitors and produce an effective iris identification method, which should be more efficacious and accurate, that is my main incitement in this research study. Furthermore, my biggest incitement lies in the conquests that iris identification has gained and attained in diverse crucial application areas such as homeland security, border surveillance and security, web security, national identity management, rapid passengers information management, controlled access to privileged information, forensic investigation and welfare management to mention but a few [12]. Hence, the vast area of applicability of the iris biometric systems suggests that the proposed iris identification would be

exceedingly useful and helpful to plenty of organizations, governments, and personal tasks. The major challenges of iris identification biometrics are:

1.2.1 Lack of Robustness

The high identification and recognition accuracy recorded by most present iris identification algorithms in highly constrained imaging conditions can be easily depleted when the same algorithm is performed in an unconstrained situation [17]. When an imaging situation is strictly controlled and a high-quality iris image is captured using advanced equipment, it becomes easy to achieve impressive results with a low error rate. [17], [18]. Though many of the latest spick-and-span iris identification methods flunk when performed in fewer cooperative iris-taken environments, in which the probability of capturing low-quality iris images is very high. In an uncontrolled imaging environment, iris images are captured with a flexible imagery setup [18]. Consequently, these images are often subject to various types of noise such as motion blur, camera diffusion, head rotation, gaze direction, camera angles, reflections, poor contrast, luminosity, occlusions, and pupil dilation. These non-idealities extremely decrease the performances of both the iris segmentation and feature extraction algorithms considerably, and as such the overall performance of the iris identification algorithm is besmirched [19].

1.2.2 Non-Circularity of the Iris Boundaries

Non-Circularity of the iris boundaries: Researchers have found that, the iris and pupil boundaries are of arbitrary shapes [10], [20]. Therefore, this can conduce to segmentation errors, if fitted with some presumed simple shapes like circles or ellipses. The accurate determination of the boundaries of the iris remains an issue until date [21].

1.2.3 Speed

Lately, the time effectiveness of available iris identification systems confines and limits the application of iris biometric systems in today's highly demanding and fast systems. Iris segmentation is supposed the bottleneck in the iris identification systems for being the greatest time-consuming module in the system [21], [11].

1.2.4 Dimensionality

The first step of iris identification biometrics focuses on developing a robust iris preprocessing algorithm aimed at handling the different challenges of non-ideal iris images captured in less-constrained environments [22]. The objective of iris preprocessing is to reduce and remove noise from iris images, augment iris features, and alleviate the information in order to obtain the desired images with a high-quality and steady region. In addition, if a full face is given in the initial step, it is significant to detect the eye first. Image filtering can be the initial and significant step in iris image preprocessing, which helps to omit intrusion and interference in the original iris image for edge detection. The important goal of an image preprocessing filter involves Noise Reduction, Color Normalization, and Edge Detection. Then a model is developed to effectively define the accurate boundaries of the iris with significantly improved speed which is a significant step in the entire iris identification procedure and it is a very critical stage as well, if the iris region is not accurately segmented, the eyelashes, eyelids, reflection, and pupil noises would present in the normalized iris region [22]. At the segmentation stage, the image is essential to be cautiously localized in order for the inner and outer boundaries of an iris can be modeled as a circle [22].

Subsequently, consideration is possessed prospering the iris identification implementation due to extracting the most significant iris textural data using a multi-scale, multi-dimensional approach based on Histogram of Oriented Gradient (HOG), VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet and InceptionV3 which are fully convolutional and are essential in improving the speed of the iris identification that can diminish the dimensionality of the iris feature vector and progress the accuracy and speed. Furthermore, in the pattern matching part, the patterns are produced at the iris feature extraction section to measure the similarity between two images of iris patterns. This stage measures the similarity and dissimilarity between the two binary codes for making the decision of acceptance or rejection [23]. In this research, for iris pattern matching and iris image matching the enhanced ORB algorithm is proposed.

1.3 Iris Biometrics Overview

Iris identification biometric system nowadays probably can be the most reliable and consistent biometric technique for person authentication which lately has been successfully deployed in many large-scale biometric-based identity management systems where the accurate authentication of a person's identity is a critical issue [21]. Such large-scale authentication systems consist of: authentication of users demanding access to resources in a networked computer environment, border control systems in airports, boarding of commercial flights, accessing home appliances remotely, performing bank transactions, gaining access to nuclear facilities, automated teller machines (ATMs), laboratory and etc. [24]. There are numerous vital characteristics of the human iris that made the iris tremendous and could be used for biometric traits and those features are its non-invasiveness, the uniqueness of iris textures, the stability of iris patterns, the public acceptance of iris biometrics, and the availability of user-friendly capturing devices to mention, these exclusivities of features have fascinated researchers and scientists in this evolving field over the past decade [25].

The iris is the annular part of the human eye which is positioned between the dark pupil and the whitish sclera see Figure 1.2 [26]. The human iris is made up of many astonishing structures that provide many interlacing minutiae and elaboration characteristics such as coronas, freckles, stripes, crypts, collarette, ciliary, radial and concentric furrows and the pupillary area, which are unique to each individual person, and to even each eye [27]. The human iris is an internal organ that is optical while visible outwardly, the iris uses the contraction and dilatation of the sphincter muscles in order to control the amount of light that enters the eye through the pupil [14]. The ventilatory filamentary tissues that make up the iris structure are very complex and very unique to each eye and individual [14], [21]. The iris pattern is generated by a chaotic process that is extremely independent of the individual's genetic structure [21]. The formation and construction of the human iris begin at the primary stage of conception, about the first three months, and the iris structures are completely formed by the eighth month of conception [21], [11]. Even though the color and pigmentation of the iris might continue to build throughout the first year of a child's birth [11], the structure

of the iris remains stable throughout a person's lifetime, except for direct physical damage or changes caused through eye surgery. In recent research works, the stability of the human iris for a lifetime has been questioned, but no credible experimental proof has been provided yet to refute the claim that iris structures remain stable for a lifetime [16]. The individuality and unification of iris patterns in addition to the further advantage that the iris is an internal organ thus far visible externally make it less susceptible to damage over a person's lifetime [14]. The necessity of the eye as a visual organ will deter many criminals from taking drastic measures to try to change their iris identity through other means, Figure 1.3 and Figure 1.4 demonstrates the eye and iris anatomies, respectively, the iris anatomy is supplementary and relevant to the proposed iris identification methodologies [21].

1.3.1 Features of the Iris

As discussed previously that the iris is made up of intricate characteristics that can be the cause for its uniqueness, individuality, and reliability. A few of those characteristics are considered below:

- I. Medial canthus: The angle formed by the upper and lower eyelids at the midpoint of the face [21].
- II. Sclera: The white area surrounding the iris in an image of the eye is typically referred to as the sclera [21].
- III. Pupil: The iris typically surrounds the darkest portion at the center of an eye image [21].
- IV. Pupillary Area: The inner part of the iris is where the pupil is located and is surrounded by sphincter muscles that define its shape [21].
- V. Ciliary Area: This area, known as the iris region, spans from the pupillary area to the ciliary body. It includes the dilator muscles that control the opening and closing of the pupil [21].
- VI. Stroma Fibres: The visible patterns of the iris are mostly made up of pigmented

fibrovascular tissue. There are two types of inconsistencies commonly found in the distribution of stroma fibers: crypts and furrows [21].

VII. Collarette: This refers to the boundary between the pupillary and ciliary areas [21].

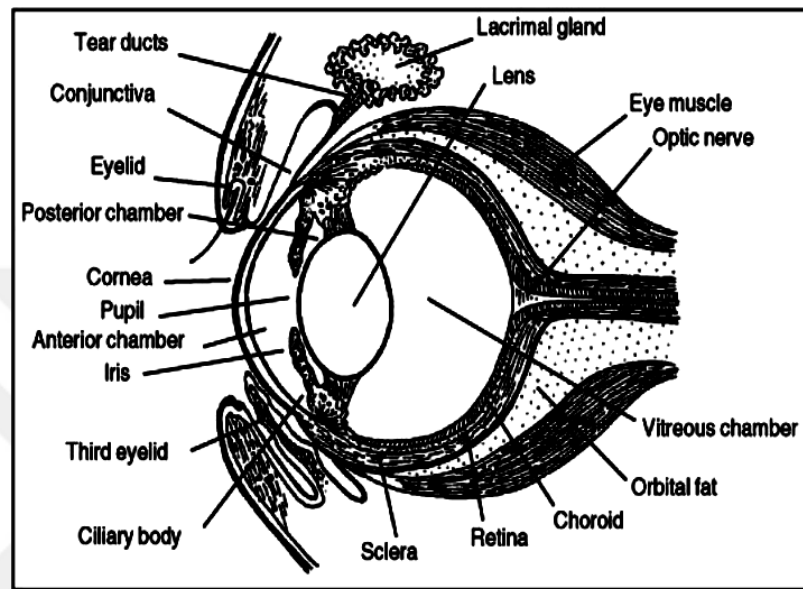


Figure 1.3 The anatomy of the human eye [21]

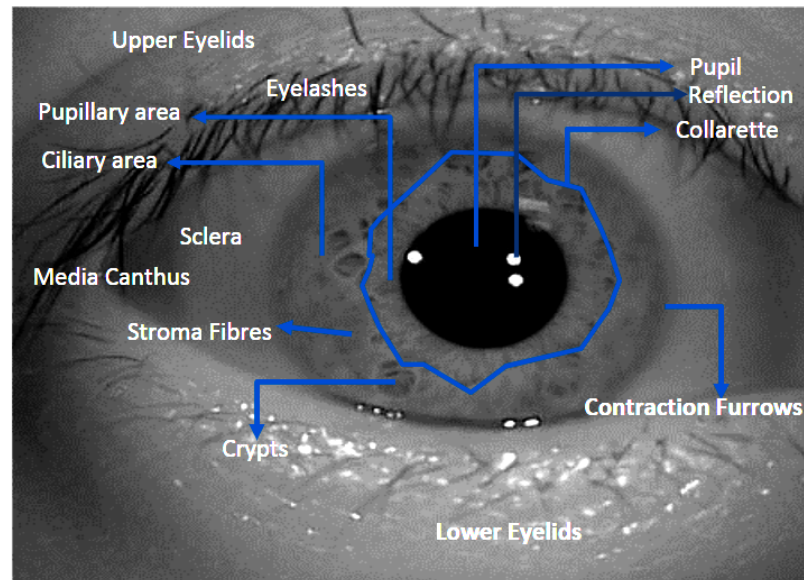


Figure 1.4 The anatomy of the human eye [21]

1.4 The proposed Iris Identification

Figure 1.5 demonstrates the block diagram of the proposed iris identification that is made up of two essential building blocks: the primary one is the iris image acquisition block and after then the iris identification steps. The iris identification block has made up of five distinct steps containing the iris image preprocessing, iris image segmentation, iris image normalization, iris feature extraction, and finally iris pattern matching and iris classification. Iris Image Acquisition step: in any iris identification, iris image acquisition can be the initial step. As well as image acquisition could be defined as the performance of obtaining an image or images from sources. This step is figured out as the critical and serious part of the iris identification system meanwhile all the other subsequent phases depend extremely on the iris image attributes obtained in this initial step. Most traditional iris identification and recognition systems do not include this step but it is significant to have in mind when designing an iris identification system, A particularly designed camera is used to capture a sequence of iris images. An iris image-capturing device considers the following key factors [18]:

- i. The lighting of the system [18].
- ii. The positioning of the system [18].
- iii. The physical capture system [18].

Iris Identification Step: The efficiency and success of the whole iris identification is extremely appertain in this module. This phase contains five different phases: iris image preprocessing, iris image segmentation, iris image normalization, iris feature extraction, and finally pattern matching and classification. In the first step, it is essential to reduce and omit the noise from the iris images, augment information, and clarify and cut down the data in order to obtain an ideal iris image with a steady region and high quality. In addition, if a full face is given, it is significant to detect the eye. Furthermore, in the second phase, the portion of the eye image that belongs to the iris is segmented from the rest of the image, and the segmented iris region is unwrapped into a set of rectangular block sizes in order to avoid size inconsistencies [22]. On the off chance that the iris locale isn't precisely portioned, the eyelashes, eyelids, reflection, and understudy

noises would display within the normalized iris locale. This is often taken after the extraction of the foremost separating highlights from the unwrapped picture and at last, the extricated highlights are used for design coordinating and classification with put away iris layouts within the iris database for personal confirmation.

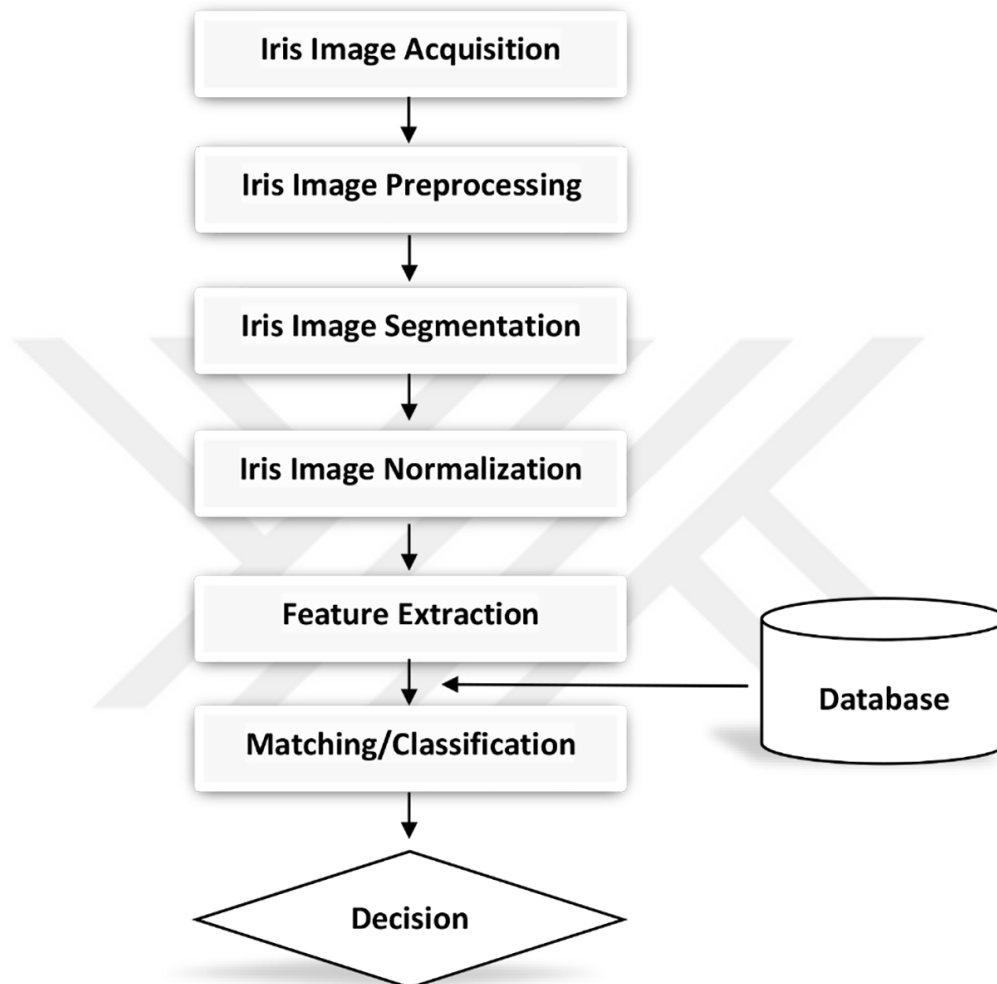


Figure 1.5 Iris Identification stages

1.5 Strengths and Weaknesses of the Iris Identification Biometric

Among all types of biometric traits, the iris biometric has been considered the most highly accurate and reliable biometric trait [14], [28]. However, it has a few inherent pitfalls that cannot be overlooked. The major qualities and shortcomings of the iris as a biometric strategy are examined separately in Sections 1.5.1 and 1.5.2.

1.5.1 Strengths of Iris Biometrics

- The iris patterns have small intra-class variability [29] [30].

- The iris is a well-protected internal organ of the eye that contains a high degree of randomness [12], [29].
- The iris is externally visible which makes iris image acquisition possible from a distance [30].
- The iris pattern remains steady throughout the lifetime of a person and it is assumed that each individual has a unique iris pattern [31].
- It is possible to encode the iris pattern and the identification system's decidability is tractable [14].
- No evidence of genetic influence has been found in the structure of the iris [32]. Consequently, the iris structures in both eyes of the same person are different and those of identical twins are also different [33], [34].
- Iris identification systems necessitate very low maintenance costs with high interoperability between different hardware vendors. Also, iris identification technology has the ability to work well with applications from other vendors [16], [35].

1.5.2 Weaknesses of Iris Biometrics

- It is arduous to capture the iris image since the size of the iris is very small (its approximate diameter is 1 cm) or the distance is too much. A specialized camera with an immense apparatus setup is needed to acquire iris images [30].
- The iris could be partially obstructed by lower and upper eyelids, and adumbrate by eyelashes, reflections, and lenses [18], [31].
- The size of the pupil changes, non-elastic deformation is a major drawback [30], [31].

1.6 Objectives and Aims of the Research

The purpose of this research is to examine the identification and techniques of iris, covering the technical background, concepts, ideas, and current practical issues related

to them. The significance would be to create modern strategies and approaches for moving forward the exactness and diminishing the preparing misfortune, as well as time proficiency of existing iris identification frameworks. The primary research includes many steps relating to the study of iris image preprocessing, iris segmentation, feature extraction, pattern matching, and classification, which emphasize improving speed and accuracy. The conceivable alterations in iris identification systems were discussed previously and in concurrence, the Gaussian filter is used in iris image preprocessing for the purpose of Noise Reduction, Color Normalization, Edge Detection, and Histogram Equalization, additionally, filters are primarily used to support the high frequencies in the image or the low frequencies in the image.

Hough transform which can conveniently detect the boundary of the pupil, Canny edge detection which is utilized to boost the iris outer boundary that is not identified competently in ordinary situations, Sobel edge detection and ellipse fitting were introduced for iris image segmentation, iris image normalization, feature extraction, finally pattern matching and classification of the iris which are fully convolutional. The research attempts to demonstrate that accurate, robust, fast, and implementable iris recognition is a viable proposition. This research endeavors to demonstrate and illustrate that precise, strong, quick, and implementable iris identification could be a practical movement. More particularly, the research destinations and aims incorporate the following:

1. To create a viable reflection discovery and evacuation strategy which is able to protect the structure of the unaffected parts of the iris image as much as conceivable for advanced processing.
2. To create a high-performance inner and outer iris boundary segmentation technique that is vigorous sufficient to handle commotion and occlusions in iris images and able to bargain with the shape inconsistencies of the iris boundaries.
3. To contemplate and execute a modern iris features extraction and completely convolutional, which is able to extract and represent the iris test with diminished dimensionality iris layout while keeping up high accuracy and performance.

4. To analyze and assess a range of iris identification frameworks applied to iris images, in order to recognize the preferences and drawbacks offered by the different approaches.
5. To assess the proposed iris identification and display the result in a standard format.
6. To distinguish the confinements of the ultimately proposed iris identification and propose a line of advanced investigation to examine these impediments.

1.7 Structure of the Thesis

The research presents iris identification based on deep convolutional neural networks in an exertion to attain the aim and targets laid out within the previous area. The structure and contents of this research are explained as takes after on a chapter-by-chapter premise.

Chapter 2: Gives a background to the field of biometrics and iris identification and a brief comprehension of past works within the field.

Chapter 3: Will discusses the methodology and iris identification stages, based on previous discussion, the proposed iris identification system includes iris image acquisition, iris image preprocessing, iris image segmentation, iris image normalization, iris feature extraction, and finally iris pattern matching and classification. Figure 2 visually describes the proposed iris identification system.

Chapter 4: Analyses and the results of the propped research of all stages in iris identification, in which the implementation of the proposed approaches was conducted utilizing three datasets that incorporate more than 2000 iris images captured beneath changing natural conditions. Additionally, it will present the proposed models in comparison with other existing progressed approaches to assess the performance.

Chapter 5: Examines and concludes the general comes about and commitments of this work. The chapter closes with a few pointers and comments on the longer-term work determined by the research.

CHAPTER 2

Background Study and Literature Review

This chapter provides background information on some of the research methodologies used and an overview of some key works that are linked to the research. In the background study, several crucial methods employed in this research, including Deep Learning, Convolutional Neural Networks, Neural Networks, Gaussian filter, Hough transform and Canny edge detection, Sobel edge detection, Ellipse Fitting, U-Net model, the Homogeneous Rubber Sheet Model which contrived by Daugman, Histogram of Oriented Gradient (HOG), VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet, and InceptionV3 and ORB algorithm are explained and discussed.

The literature review summarizes and compares a few of the well-known strategies utilized in iris identification systems which are at the cutting edge of this energizing and challenging biometric field. A brief review of some tenacious methods and techniques utilized in iris image preprocessing, iris image segmentation, and iris image feature extraction are also presented and discussed.

2.1 Deep Convolutional Neural Network Background

Also known as ConvNets, DCNN, or CNNs. It is usually used in image recognition or classification. This technique developed in the 1980s - 1990s but it revived in 2012 and began to be used in most of the fields of computer vision and has begun to grow fast [36]. DCNN does not just contain many hidden layers, but it simulates the brain by identifying images, it has the ability to extract features instead of manually designing them by using a number of locally connected layers that have the ability to automatic feature recognition, and a number of fully connected layers for classification [36].

The feature extraction of neural networks consists of special types of neural networks, determined by training and updating weights, DCNN converts the manual methods for extracting features into an automatic process [37]. DCNN consists of a set of layers each

layer performs a specific function [37]. The most common layers in terms of structure and function will be illustrated, the convolutional layer (conv): is made from sets of learnable filters, and each filter consists of a set of weights [38]. The weight values were selected randomly and learned via a backpropagation algorithm; the learned weight is the filter that convolved through the whole image where the result is the feature map [38]. The feature map identifies the distinct features present in the input image. This map is calculated using Equation 2.1 [39]:

$$Z^s = f\left(\sum_{t=1}^q W_j^s * X^i + b_s\right) \quad (2.1)$$

Where $*$ is a two-dimensional discrete convolution operator, b is a trainable bias parameter and f is the activation function, the Rectified Linear function (ReLU) is the common activation function used with DCNN [39].

Pooling layer (pool): Max-pooling layers compute the maximum value of a local patch (2×2 or 3×3) of the convolutional feature map output units, so, it reduces the dimension of the feature representation [40]. The main advantage of using a pooling layer is to reduce memory requirements and computational time by removing the features that are not important in CNN. The max pooling operation can be represented in Equation 2.2 [39] [40] :

$$y_{j,k}^i = \max_{0 \leq m,n < s} (x_{j,k}^i + m, k.s + n) \quad (2.2)$$

Where $y_{(j,k)}^i$ stands for a neuron that is found in the i^{th} output activation map and it is calculated across an $(s \times s)$ non-overlapped local area in the i^{th} input map $(x)_{(j,k)}^i$, fully connected layer (FC): the output of the last pooling or convolutional layers is input to fully connected layers (FC) same as a traditional neural network, softmax layer: the last layer of a DCNN that it uses to classify the extracted feature from previous layers into N classes, it is often outputting the class membership probabilities that turn numbers from logits into probabilities that sum to one. The softmax operation can be calculated in Equation 2.3 [39] [40] :

$$softmax(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (2.3)$$

2.2 Iris as Biometric Traits

The iris is the annular-shaped part of the eye between the pupil and the sclera which regulates the amount of light entering the eye through the pupil [4], [41]. The iris is physically small in size (around 11mm) but a well-designed visual system is able to magnify the human iris into a high-resolution image that is approximately 200 to 300 pixels in diameter [28], [29], [42] [43], [44]. The iris comprises many minutia features such as freckles, coronas, collarette, stripes, stroma, furrows, crypts, etc., which make up the iris patterns [41]. It must be noted that the visual pattern of the human iris includes both color and texture [42]. In any case, iris color has restricted segregation control for individual identification and acknowledgment. In this manner, gray-level iris images taken beneath near-infrared enlightenments are frequently utilized to record iris designs for individual identification and confirmation.

As previously discussed, the iris texture pattern is formed and becomes steady within the first eight months of human gestation [28]. It is regularly assumed that the arrangement of iris patterns is decided by the development environment, i.e. iris may be a phenotypic biometric characteristic. So indeed indistinguishable twins can be separated by utilizing reasonable iris highlights; indeed the iris patterns of the left and right eyes of the same individual are diverse. The primary challenge that regularly faces the automatic identification of people by their iris patterns is the assurance of the iris parcel from the eye picture which is called segmentation.

2.3 Iris Image Preprocessing

Before iris image segmentation it is essential to discuss image preprocessing in order to make the iris images present for segmentation. The objective of iris image preprocessing

is clearly to reduce and remove the noise from iris images, augment the data, and abridge the information and data in order to obtain an arbitrary iris image with a stable region and high quality. In addition, if a full face is given in the initial step, it is significant to detect the eye, which is extremely beneficial and significant in biometric systems, while Figure 2.1 exhibits eye detection from the GI4E database. To attain this purpose, filtering the images and detecting the image edge is necessary for iris image preprocessing [45].



Figure 2.1 Eye Detection image from GI4E database

2.3.1 Gaussian Filter

Gaussian Filtering [46], [47] has been intensively studied in image processing and computer vision. Image filtering is the first step in iris image preprocessing, which is to omit the intrusion and impedances within the original iris image to make it ready for edge detection. The important goal of the image preprocessing filter involves Noise Reduction, Color Normalization, Edge Detection, and Histogram Equalization. In this research Gaussian filter has been used to decrease the search space the eye image must be upgraded sometime recently and after creating the edge outline. Therefore, in this research, a 2D Gaussian filter is applied in order to smooth the eye image and reduce noise, the Gaussian filter is a low-pass filter, whose 2D filter coefficients are computed in Equation 2.4 [47]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.4)$$

Where x and y are the distances from the origin along the horizontal and vertical axes, and $\sigma = 0.8$ is the standard deviation of the Gaussian distribution [47].

2.3.2 Hough Transforms

First of all, the edges of the iris are needed to be detected utilizing filters, after then by boosting those pieces of the iris image that possess edges, when an image's sharpness enhances, then it will become more prominent. In image processing Edge detection is a technique used to find the boundaries of substances through the images, it detects interruptions in brightness.

Edge detection is commonly used in order to segment the image and extract data from the image, in areas like image preprocessing, machine vision, and computer vision. Furthermore, Edge detection can be the most important part of the iris identification method, edge detection method forms pupil boundary detection precisely and easier. The pupil is a superior fragment of the image of the eyes, Hough transforms it could be easily detected the boundary of the pupil by many methods such as Canny edge detection [48], Prewitt edge detection, the Sobel edge detection can also be used to find the pupil [49]. Hough Transform to detect the circle is computed in Equation 2.5:

$$r^2 = (x - X_c)^2 + (y - Y_c)^2 \quad (2.5)$$

For the purpose of identifying the circle in the iris image, we need to know its radius and center. The radius is represented by r , while X_c and Y_c represent the x and y coordinates of the center. All the pixels with features that are detected through the edge detection algorithm will be processed in the xy space. The Hough Transform Algorithm will be used to explain the process [48]:

- Read the image and perform the image processing necessary [48].

- Apply canny edge detector [48].
- Initialize 3-D accumulator array.
- Find radius and Increment 3-D accumulator array [48].
- Finding Peaks [48].
- Finding peak for the center of the circle.
- Finding peaks from an array of radii [48].
- As the proposed iris identification system is to find iris from the image as well, the circles looked for are concentric circles. Hence if the center of the circles is found then from the accumulator array radii of that particular center coordinates are searched and stored [48] [49].
- Using the hill-climbing technique, peaks are identified from the stored radii. The smallest peak among the stored peaks is then considered as the radius of the pupil.
- Finding RGB values at different Radii [48].
- Comparing average RGB values for authorization [49].

Perform Canny Edge Detection: In order to perform canny edge detection, the image needs to be converted into a gray-scale image from an RGB image [48]. The Canny edge detector is the first derivative of a Gaussian, it approximates the operator that optimizes the product of SNR [50]. Canny edge detection is appropriate for detecting the pupil, Canny edge detection is utilized to enhance the iris outer boundary that is not identified competently in ordinary conditions [49].

2.4 Iris Image Segmentation

Iris image segmentation is crucial for the accurate identification of individuals based on their iris patterns. This process involves identifying the relevant fragment of the eye which pertains to the iris. To achieve this, the iris image must be properly centralized

so that the outer and inner boundaries of the iris can be represented as a circle. This section presents and analyzes various techniques for accurately defining the iris region within the eye image.

2.4.1 Daugman's Differential Operator

The iris is the circular part of the eye that is located between the pupil and the sclera, Daugman [29], [51], in his early work proposed a method of approximating each boundary of the iris as circles. Therefore, each boundary of the iris can be estimated based on three parameters which include: to detect the iris boundaries, Daugman [51] proposed an integrodifferential operator that searches through an N parameter space. The operator requires the center coordinates of the circle, x_0 and y_0 , as well as the radius, r . The integrodifferential operator can be given in Equation 2.6 [51]:

$$\max(r, x_0, y_0) \left| \left(G_\sigma(r) * \frac{\partial}{\partial r} \right) \oint_{r, x_0, y_0} \frac{I_{x,y}}{2\pi r} ds \right| \quad (2.6)$$

Where $G_\sigma(r)$ signifies the smoothing function and $I(x,y)$ represents the image of the eye with (x,y) parameters corresponding to the current pixel location [51].

2.4.2 Wildes' Hough Transform

In a similar approach, Wildes [52], proposed a different procedural method for the estimation of iris boundaries. Wildes proposed the computation of binary edge maps followed by a Hough transform for the detection of circles in the iris image, it involves two steps: first, a binary edge map is generated by using a Gaussian filter, then, a Hough transform technique is employed to consider the set of edge points and find the circle that best fits the most edge points [52]. Subsequently, votes are cast in a circular Hough space and analyzed to estimate the three parameters of the circle (x_0, y_0, r) . The circular Hough space can be defined in Equation 2.7 [52]:

$$H(x_0, y_0, r) = \sum_i h(x_i, y_i, x_0, y_0, r) \quad (2.7)$$

Where (x_i, y_i) represents an edge pixel and:

$$h(x_i, y_i, x_0, y_0, r) = \begin{cases} 1, & \text{if } (x_i, y_i) \text{ is on the circle } (y_0, y_0, r); \\ 0, & \text{otherwise} \end{cases} \quad (2.8)$$

The location in (x_0, y_0, r) which has the highest value of $H(x_0, y_0, r)$ is chosen as the parameter vector for the strongest circular boundary.

2.4.3 U-Net Iris Image Segmentation

In terms of segmentation, the U-Net model has illustrated good performance, especially semantic segmentation in biomedical image segmentation [53]. such as Brain tumor segmentation in MRI images using U-Net [54], Liver and Liver Tumor Segmentation [55], Rethinking the U-Net architecture for multimodal biomedical image segmentation [56], Efficient Biomedical Image Segmentation on EdgeTPUs at Point of Care [57], and many more. The U-Net model architecture in this research with an iris image size of 240×320 pixels as an input has been used. The feature maps with (width $\times$ height) dimension along on the left represented by an individual box. Designed for Conv 1, two convolution operations by 3×3 convolutional kernels were performed, with the size of 256×256 pixels the number of feature maps is 64, and the max-pooling was done using a 2×2 kernel size. Figure 2.2 demonstrates the complete structure of U-Net segmentation used in this research for iris image segmentation, which stays a representative encoder-decoder architecture.

2.5 Iris Image Normalization

Iris images might be taken in several sizes and with different distances, as a result of brightness differences, the circular and redial size of the pupil may vary appropriately. Thus, the iris region has to be normalized to reimburse for this alteration. When the iris identification method takes an image from a person's eye. Afterward, this image needed to be segmented into different parts and then normalized for the purpose of the feature extraction process. The proficiency and implementation of the iris identification technique particularly depend on good quality segmentation and normalization. The

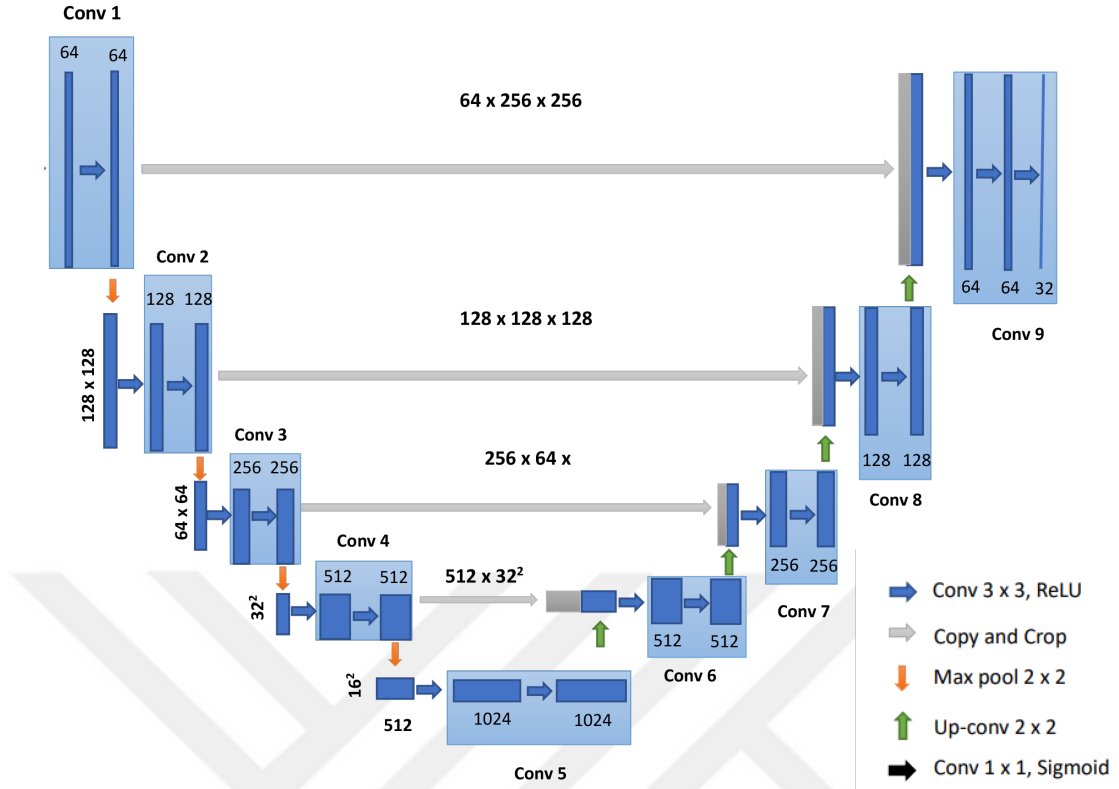


Figure 2.2 U-Net Structure used for Iris Image Segmentation

normalization process normalizes iris images to a constant dimension and rectangular shape to avoid spatial incompatibility which might arise for various reasons such as iris imaging distance variation, the revolution of the camera, illumination variation, iris stretching, eye rotation, head deformation and etc. Therefore, iris image normalization is applied to minimize all of these obstacles.

2.5.1 Homogeneous Rubber Sheet Model

The Homogeneous Rubber Sheet Model which was developed by Daugman and proposed the usage of a Rubber Sheet Model, assigns each point in the iris region as pair antithetical coordinates $(r\theta)$, where r is the distance $[0, 1]$ while θ denotes the angle $[0, 2\pi]$ [58]. The model is illustrated in Figure 2.3.

Defining the correction of the iris region from Cartesian coordinates (x, y) , to normalize nonconcentric and abnormal polar coordinates is demonstrated as follows in Equation 2.9 [48]:

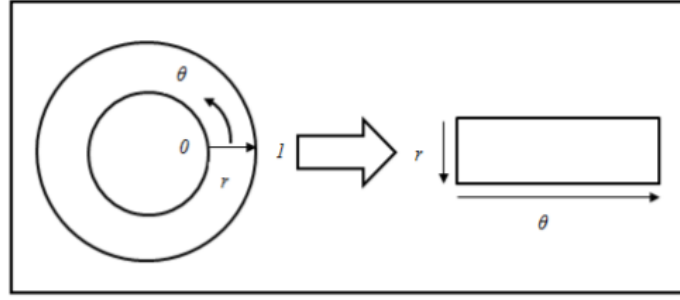


Figure 2.3 Rubber sheet model [29]

$$\begin{aligned}
 I(x(r, \theta)y(r, \theta)) &\rightarrow I(r, \theta) \\
 x(r, \theta) &= (I - r)x_p(\theta) + rx_I(\theta) \\
 y(r, \theta) &= (I - r)y_p(\theta) + ry_I(\theta)
 \end{aligned} \tag{2.9}$$

Where $I(x, y)$ is represented as the iris region (x, y) indicates the original Cartesian coordinates, (r, θ) demonstrates the corresponding normalized polar coordinates, x_p, y_p and x_I, y_I are the coordinates of the pupil and iris boundaries along θ direction [59].

2.6 Iris Feature Extraction

When the iris images have been magnificently segmented and normalized, the afterward phase is to extract considerable data from the iris pattern that has been obtained from normalized iris images. The features which are extracted from iris images are programmed to produce iris patterns. Several iris recognitions take the benefit of using bandpass decomposition of iris images to produce biometric templates [60].

There are many Algorithms used to extract image features, I can name them such as the Gabor Filter, Wavelets Transform, Laplacian of Gaussian filter, Key Local Variations Hilbert Transform, and Discrete Cosine Transform [60]. The Wavelet transform intersects the iris zone toward the synthesizer with several resolutions. Principally, the greatest commonly used wavelets are Daubechies, Biorthogonal, Haar, and Mexican Hat wavelets [60]. Compared to the advantages of Wavelet transform With Fourier transform, the Wavelet has both space and frequency resolution for

extracting important features, thus, the concluding output from the Wavelet filter is encoded and produced a biometric template [61].

2.6.1 Histogram of Oriented Gradient (HOG)

In this research first of all Histogram of Oriented Gradient (HOG) has been utilized. A Histogram of oriented gradients (HOG) can be stated as a well-accredited and suitable feature for feature extraction as a consequence of its preferable implementation and fairly modest calculation. Since first proposed in [62], mathematically, for a two-dimensional function $f(x, y)$, the gradients can be computed by the derivatives with respect to x and y and computed in Equation 2.10.

$$\begin{aligned} G_x &= I(x+1, y) - I(x-1, y) \\ G_y &= I(x, y+1) - I(x, y-1) \end{aligned} \quad (2.10)$$

Here G_x and G_y characterize the gradients in the horizontal and vertical directions individually [62]. Figure 2.4 shows the HOG iris feature extraction result.

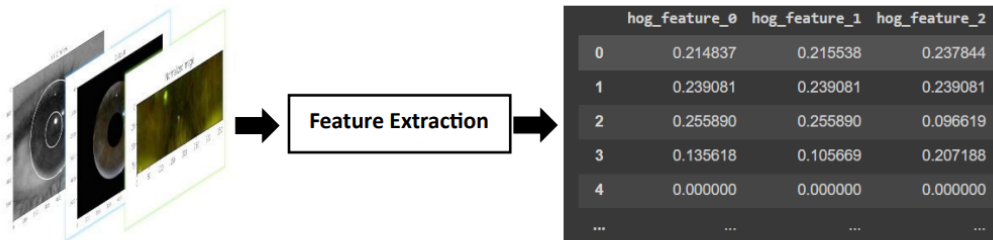


Figure 2.4 HOG Iris Feature Extraction Result

2.6.2 VGG – 16 and VGG – 19 Models

As discussed perversely, in this research, I have used several models to extract iris features additionally, I peruse the susceptibility and capabilities of a convolutional neural network comprehended using the VGG16 method, mostly known as a convolutional community model. Figure 2.5 represent the VGG-16 Iris Feature Extraction Architecture. It utilizes 16 layers with weights and is thought to as one of the best vision model architectures to date. VGG-16 is a convolutional neural network

trained on a subset of the ImageNet dataset, a collection of over 14 million images belonging to 22,000 categories, K. Simonyan and A. Zisserman [63] proposed this model in their 2015 paper. The architecture of the VGG16 Model is displayed in Figure 2.5. It comprises 16 Convolutional and Max Pooling layers, 3 Dense layers for the Fully-Connected layer, and an output layer consisting of 1,000 nodes [63].

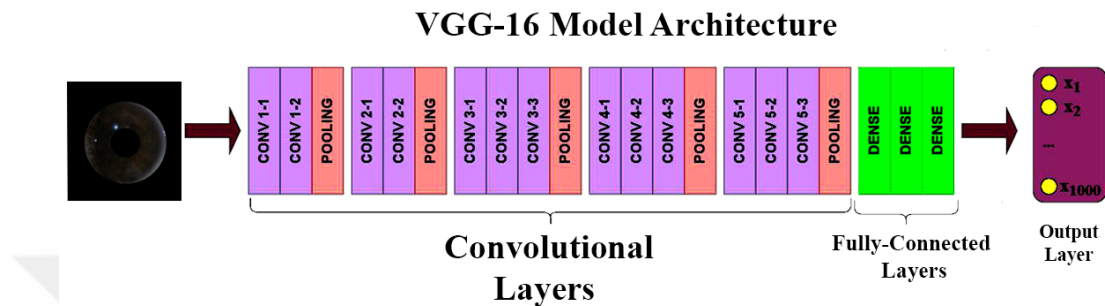


Figure 2.5 VGG-16 Model Architecture

The VGG-19 model is the modified version. The concept of the VGG-19 model (VGGNet-19) is similar to the VGG-16 except that VGG-19 contains 19 layers to be performed. In the convolutional layers, the numbers 16 and 19 indicate the number of weight layers. Respectively the difference is between the layers, VGG-19 has three extra convolutional layers compared to the VGG-16. VGG-19 is utilized as a pre-processing pre-trained model. The CNN network depth of this convolutional neural network has been enhanced, as compared to the traditional ones. By utilizing a structure that alternates between multiple convolutional layers and non-linear activation layers, this method proves to be superior to a single convolution, the layer structure is more effective at extracting image features, employing Maxpooling for downsampling, and utilizing the linear unit (ReLU) as the activation function, which involves selecting the largest value in an image area to serve as the pooled value for that area. [64]. The downsampling layer is primarily utilized to enhance the network's ability to resist distortion in images while still preserving the essential features of the sample and decreasing the number of parameters. [65].

VGG-19 is a fully convolutional neural network architecture, consisting of 19 layers of deep neural network in addition it possesses more weight. VGG-19 in input takes an image size of 224×224 , VGG's convolutional layers influence a minor 510 amendable

scope of 3×3 content to increase the depth, the minimum possible size, that is followed by a ReLU activation function. The VGG19 contains three fully connected layers [66]. Figure 2.6 demonstrates the VGG-19 Iris Feature Extraction Architecture in this work.

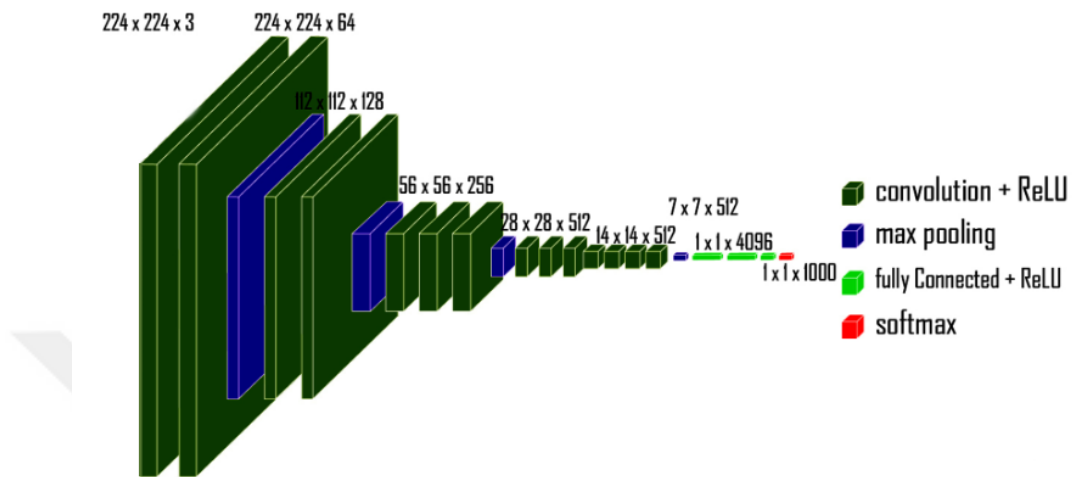


Figure 2.6 VGG-19 Iris Feature Extraction Model Architecture

2.6.3 ResNet Models

Basically, the ResNet model architecture is taken from ResNet-34 and included 34 weighted-layers, that support different ways to enhance further convolutional layers to a CNN. The established network originally was founded on the VGG neural networks, as mentioned previously each convolutional network has a 3×3 filter. Though, a ResNet has fewer filters and is less complicated than a VGG-Net. ResNet-50 is fully convolutional and contains 50 layers deep, the 50 layers in ResNet use a bottleneck meant for the building block, which decreases the number of parameters and matrix multiplications, this characteristic empowers greatly faster training of each layer [67]. ResNet-101 is the same as ResNet-50, a fully convolutional neural network but contains 101 layers deep. Figure 2.7 describe the Distribution of pre-trained CNN Models.

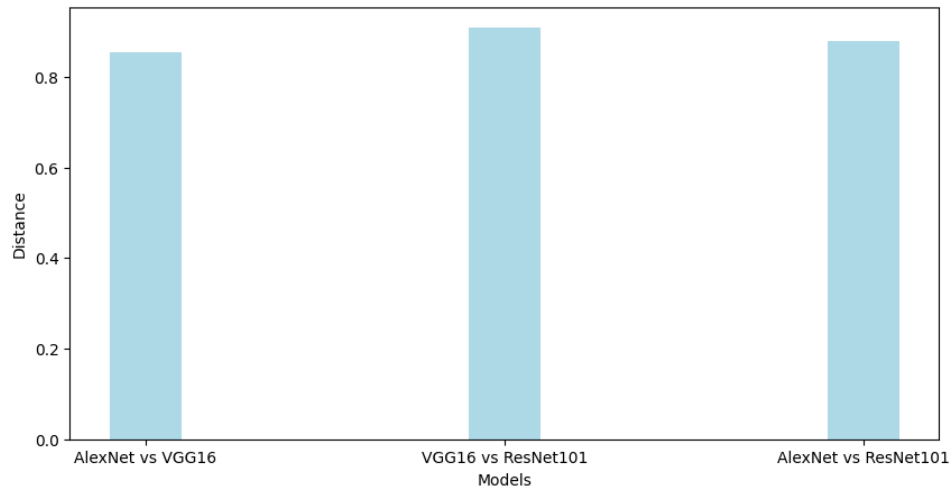


Figure 2.7 Distribution of pre-trained CNN Models

2.7 Iris Pattern Matching

The process of calculating the rating of resemblance amongst the input iris images such as test set and training set from the database and purposes to extract the most perceptive features used to produce the significant iris mystery. In the matching stage, the patterns are generated at the iris feature extraction stage to measure the similarity between two images of iris patterns, this stage measures the similarity and dissimilarity between the two binary codes for making the decision of acceptance or rejection [68].

2.7.1 ORB Algorithm

In this research, for iris pattern matching, the ORB algorithm is utilized as the matching technique and method. On this basis, the enhanced ORB algorithm is suggested. Image matching is used to find the similarities between the original mask and the mask which is predicted. The ORB image-matching algorithm is normally divided into three stages: feature point extraction, generating feature point descriptors, and feature point matching. [69] The flow chart is exposed in Figure 2.8.

- (a) Feature point extraction: The ORB algorithm uses the improved FAST (features from accelerated segment test) [69] algorithms in order to detect feature points.
- (b) Generate feature point descriptors: Then after extracting the Oriented FAST feature points, the ORB algorithm applies the enhanced BRIEF algorithm [70] to

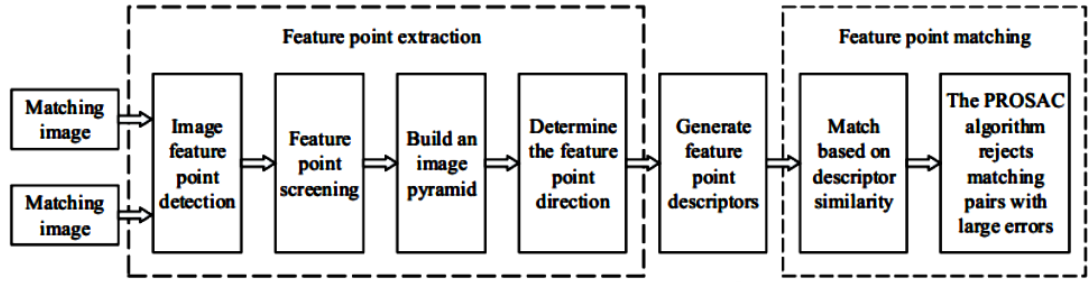


Figure 2.8 Distribution of pre-trained CNN Models [69]

calculate the descriptors for separate points. BRIEF is a binary vector descriptor, in which the vector contains a number of 0 and 1 [71], Equation 2.11

$$\tau(p; x, y) = \begin{cases} 1, p(x) < p(y) \\ 0, p(x) \geq p(y) \end{cases} \quad (2.11)$$

Where $p(x)$ can be the gray value at the ground x about the image feature point, and $p(y)$ stays the gray value at the ground y about the image feature point. [71].

- (c) Feature point matching: After the first and the second steps it is determining the scale and rotation statistics of the image feature points, as it is essential to specify the similarity between the feature point descriptors in the two different time images to determine whether they match or not [71], [72].

2.8 Different Biometric Technologies

Biometric techniques are increasingly presented as a way to fight against fraud and theft [73]. Biometrics is an automated method for identifying a person or verifying the identity of a person based on a physiological or behavioral characteristic that has the capability to reliably distinguish between an authorized person and an imposter [25]. Biometric features are distinct, cannot be ignored or missed, and require the individual to be present for identification. This makes biometrics further trustworthy and more effective than traditional knowledge-based or token-based techniques. The use of biometrics for human identification provides several inimitable benefits. Your unique biometric data can be utilized to accurately identify you. The field of biometrics offers the potential for speedy, user-friendly, precise, dependable, and cost-effective

authentication across numerous applications. This article presents an introduction to the various biometric methods, outlining their respective pros and cons.

Biometrics, which denotes identifying an individual based on his or her physiological or behavioral characteristics. Physiological characteristics include hand or finger images, facial characteristics, and iris recognition [24]. Behavioral characteristics are traits that are developed through learning and experience, dynamic signature verification, voice verification, and keystroke dynamics. [74]. It is important to note that there is no one biometric that can be considered "perfect" for all purposes. Every biometric technique may contain its own set of privileges and invalidates. However, there can be some shared traits that are essential to ensure a biometric procedure is effective. Initially, the biometric system has to be based on an identifiable trait. Scientific evidence overwhelmingly supports the notion that each person's fingerprints are unique and distinguishable from one another." Hand geometry technology has been in use for several years, while face and iris recognition technologies have become more prevalent. There are new biometric methods that may be as accurate as traditional ones, but they need more research to confirm their uniqueness, it is also important to consider how easy a system is to use [75].

Biometric authentication offers the benefit of requiring additional instances of authentication without causing inconvenience to users. As the use of biometric technologies becomes more widespread, managing multiple levels or instances of authentication will become easier for users. Biometric authentication requires comparing a registered or enrolled biometric sample (biometric template or identifier) against a newly captured biometric sample (for example, a fingerprint captured during a login), during Enrollment, a sample of the biometric trait is captured, processed by a computer, and stored for later comparison [76]. Biometric system for identification has two methods of operation: Identification method and Verification method. In the Identification method, the system identifies a single person by searching the entire enrolled population for a match based on their biometrics. In Verification mode, the system authenticates a person's claimed identity by comparing it to their previously enrolled pattern. This is also called "one-to-one" matching [77].

2.8.1 Face Recognition

There are various methods for identifying a person through their facial image. One way is by taking a picture of their face using a low-cost camera. Another technique is to utilize infrared emission patterns on the face.

Facial recognition in visible light typically models key features from the central portion of a facial image, using a wide assortment of cameras, the visible light systems extract features from the captured image(s) that do not change over time while avoiding superficial features such as facial expressions or hair [78]. Several approaches to modeling facial images in the visible spectrum are Principal Component Analysis, Local Feature Analysis, neural networks, elastic graph theory, and multi-resolution analysis, some of the challenges of facial recognition in the visual spectrum include reducing the impact of variable lighting and detecting a mask or photograph. Certain facial recognition systems may necessitate a still or positioned user to successfully capture the image, while several systems employ a live process to identify the person's head and automatically locate the face, the major benefits of facial recognition are that it is non-intrusive, hands-free, continuous, and accepted by most users [78].

Over three decades of research and development in face recognition, high performance has been achieved in many applications, nevertheless, the recognition methods are still affected by factors such as illumination, facial expression, and poses, as described in [79]. Efficient attempts to find solutions to these problems have been made with Machine Learning (ML) as well as with Deep Learning (DL) methods and techniques which extract hierarchical and semantic structures existing in images [80]. In most face recognition technologies features extracted from images are associated with the closest dictionary term using the Nearest Neighbors or similar techniques in a feature parameters space, then matches between the image and the dictionary features are normalized and evaluated, and the result is then a vector representation of the given j -th image I_j in the dictionary feature space as follows in Equation 2.12 [79] [80]:

$$I_j = a_{1j}v_1, a_{2j}v_2, \dots, a_{nj}v_n, \quad (2.12)$$

where v_i are the feature dictionary basis vectors, and a_{ij} are the numbers of occurrences in the I_j image normalized by the count number of all features in the image [80]. The SURF algorithm utilizes the integral image [79] to prevent the need for recalculating sub-sampling at different scales. This is achieved by summing up the intensities of all pixels within a box at the current point, allowing for a constant speed retrieval of rectangular area intensity. Box filters that approximate Gaussian filters are utilized. Instead. Following [81], the scale-normalized determinant of the Hessian matrix is used to estimate the second-order gradient Equation 2.13:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy}), \quad (2.13)$$

where D_{xx} is the box filter approximation of the second order Gaussian derivative at the i -th pixel location $x_i = (x_i, y_i)$ with the scale σ of the image I , the above feature representation phase is built on feature descriptors that characterize the neighborhood of each of the interest points in all the scales and their dominant directions, the convolution of the Gaussian filter and the first-order gradient measures in various directions for the neighboring grid components comprise the feature space basis [82].

In particular, the SURF algorithm estimates the Gaussian and Haar wavelet convolution using box filters in the neighborhood radius defined by the scale in the number of rotational steps, the orientation that yields the highest linear sum of the various Haar components is selected as the dominant one [80]. Finally, in the chosen direction a 64-dimensional descriptor is built using the Haar components and their absolute values, calculated for each 4×4 cell of t -th region of the image I , see Equation 2.14 [80]:

$$v_t = \left(\dots, \sum d_{xi}, \sum d_{yi}, \sum |d_{xi}|, \sum |d_{yi}| \dots \right) \quad (2.14)$$

where $\sum d_{xi}$ is the sum of the Haar wavelet details responses in the horizontal direction of pixels in the i -th cell of the region, $i \in 1, \dots, 16$, and $\sum |d_{xi}|$ is the sum of absolute values of the wavelet which was responded, and y denoted the vertical direction of the Haar wavelet responded, as described in [80] [82].

Additionally, Artificial Neural Networks (ANNs) have been shown efficient in learning

feature representation [83], [82], for example, an ANN has been used for generalized multivariate piecewise regression f in Equation 2.15:

$$f : X \subset R^m \mapsto Z \subset R^n \quad (2.15)$$

In particular cases, an ANN implements a multivariate linear regression [82]:

$$y = f(X) \mathbf{W} \mathbf{x} \quad \forall \mathbf{x} \in X \subset R^m \quad \forall x \in \mathbf{Y} \subset R^n \quad (2.16)$$

where $\mathbf{W} \in W \subset R^n \times R^m$ is the adjustable coefficient matrix. ANN with more than one hidden layer is called Deep Neural Networks (DNN), and corresponding ML methods are called Deep Learning (DL) [82]. DNNs implement a method of the substantiated that has numerous learning parameters. These parameters map ANN layers of various architectures, resulting in a more accurate emulation of reality. A non-linear activation function is typically introduced to achieve this.

2.8.2 Voice Recognition

Voice recognition technology has been around for about forty years. It works by averaging the output of various analog filters over time to identify a match. This technology uses the unique acoustic features of speech that vary from person to person, these acoustic patterns reflect both anatomy (e.g., size and shape of the throat and mouth) and learned behavioral patterns (e.g., voice pitch, speaking style) [74]. The process of incorporating learned patterns into voice templates, also known as "voiceprints," is known as speaker recognition and has been classified as a "behavioral biometric", there are three styles of spoken input that are used in voice recognition systems: text-dependent, text-prompted, and text-independent [84]. Voice verification applications typically use text-dependent input, where users select and enroll one or more voice passwords [74]. If there are concerns about imposters, text-prompted input may be used. Voiceprints are processed and stored using a range of technologies, including hidden Markov models, pattern-matching algorithms, neural networks, matrix representation, and decision trees. However, changes in the voice's behavioral attributes or using different phones for enrollment and verification may lead to

performance degradation. Recognition systems need to account for voice changes due to aging as well. Numerous companies advertise voice recognition engines, frequently as components of extensive voice processing, control, and switching systems. Capture of the biometric is seen as non-invasive. The technology needs little additional hardware by using existing microphones and voice-transmission technology allowing recognition over long distances via ordinary telephones (wireline or wireless) [84].

A Deep Neural Network (DNN) is trained by optimized features [85], and a genetic algorithm is used to determine the weights and biases of the neural network. The fitness function of a genetic algorithm can be defined according to the requirement. In [86], f_s is the currently selected feature, and it is the threshold value of feature points, on the basis of the given condition, they checked the fit value that could exist in the new feature set, as computed in Equation 2.17.

$$f(fit) = \begin{cases} 1, & f_s < f_t \\ 0, & f_s \geq f_t \end{cases} \quad (2.17)$$

Where $f(fit)$ is the fit value according to the fitness function [86]. Neural Network Toolbox supports a variety of supervised and unsupervised architectures for training in pattern recognition.

The main goal of an Automatic speech recognition system is to transform an audio input signal $x = (x_1, x_2, \dots, x_T)$ with a specific length T into a sequence of words or characters (i.e., labels) $y = (y_1, y_2, \dots, y_N)$, $y_N \in \mathbf{V}$ where $\mathbf{V}$ is the vocabulary, the labels may either be character-level labels, that refer to (i.e., letters) or word-level labels, which refer to the whole (i.e., words) [87]. The most probable output sequence is given as follows where $y \in \mathbf{V}$.

$$\hat{y} = \operatorname{argmax} p(y|x) \quad (2.18)$$

When it comes to perceiving audio signals, the human ear operates in a nonlinear manner. To adjust to variations in pitch, the Mel scale has been invented to create a straight representation of how the human ear hears sounds. Frequencies within the range of 0 – 1kHz can be converted to the Mel-scale, while all other frequencies are

logarithmic. The Mel-scale frequency is computed in Equation 2.19 [81] [85]:

$$f_{mel} = \frac{1000}{\log(2)} \left[1 + \frac{f_{Hz}}{1000} \right] \quad (2.19)$$

where f_{Hz} is the frequency of the original signal.

Recurrent Neural Networks process information based on the sequence of time, their current hidden state depends on all of their previous hidden states. These models are specifically created to represent time-series signals and identify both short and long-term dependencies among various input time steps. This is particularly relevant for speech recognition purposes, the input signal $x = (x_1, x_2, \dots, x_T)$ is passed through the RNN to compute the hidden sequences $x = (h_1, h_2, \dots, h_N)$ and the output sequences $y = (y_1, y_2, \dots, y_N)$ respectively. One major drawback of the simple form of RNNs is that it generates the next output based only on the previous context [85].

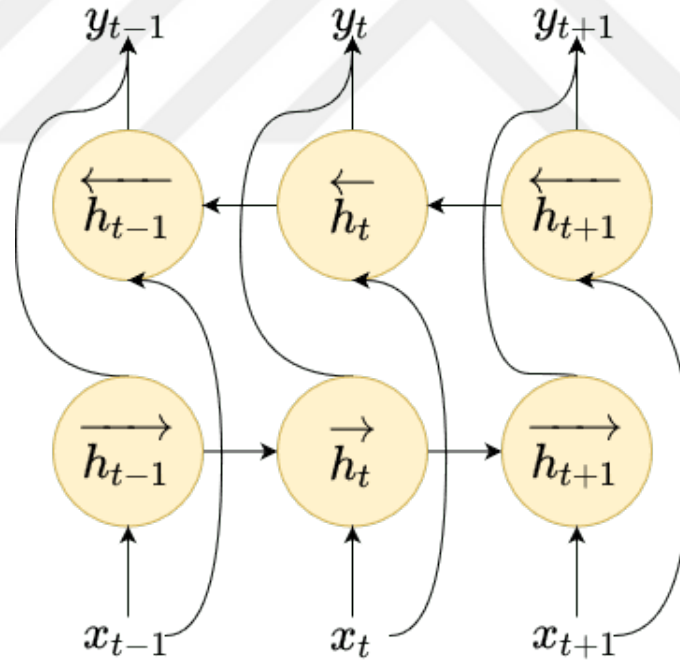


Figure 2.9 Bidirectional RNN[87]

RNNs compute the sequence of hidden vectors h in Equation 2.20 [85]:

$$\begin{aligned} h_t &= H(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \\ y_t &= W_{hy} h_t + b_y, \end{aligned} \quad (2.20)$$

where W are the weights, b are the bias vectors and H is the nonlinear function [85].

2.8.3 Hand Geometry and Finger Print

These methods of personal authentication are well established. Hand recognition technology has been in existence for more than two decades. To achieve personal authentication, a system may measure either the physical characteristics of hands and fingers including their length, width, thickness, and surface area [88]. An intriguing feature of certain systems is their need for a tiny biometric sample, typically just a few bytes. The use of hand geometry has become widely accepted for various applications. It can frequently be found in physical access control in commercial and residential applications, in time and attendance systems, and in general personal authentication applications [88].

There are many kinds of biometric traits, and in general, they are grouped into two categories: physical and behavioral traits. While physical traits include fingerprint, face, and iris; behavioral traits include gait, signature, and writing behavior. Except for some anomalies and disabilities, everyone has fingers and thus fingerprints. Ridges and valleys on the fingers can be easily captured by using various imaging techniques and sensors. Due to the sweat and the oil surface on the finger skin, they are left on the surfaces and can even be collected from where they make physical contact [89]. In fact, automated fingerprint systems have been actively used for decades now. However, more robust and accurate systems are always needed because fingerprint recognition is still a challenging pattern recognition problem with high intra-class variability and high inter-class similarity [90].

Deep learning is proven to perform well on many pattern recognition and computer vision tasks. Since fingerprint analysis is a sub-domain of computer vision and is actually a pattern recognition problem [89]. Utilizing deep learning can enhance the accuracy and robustness of fingerprint analysis with minimal effort or even no need for intermediate tasks like image enhancement and feature extraction. In the past, different machine-learning methods have been used for different parts of fingerprint recognition. However, those require extra pre-processing effort prior to using machine

learning algorithms for the task at hand. For instance, for fingerprint classification and minutiae extraction, orientation fields should be extracted and only then they are fed to a machine learning algorithm for better performance. In contrast, when deep learning is adopted, the network is expected to automatically learn the important features from the data and use them to make necessary decisions [89].

Many approaches have been developed to classify fingerprints and in most of them, fingerprint classification is divided into two main tasks: feature extraction and classification. The features mostly used for fingerprint classification are global features [90]. In Figure 2.10, most of the techniques concentrate on ridge-line flow, orientation image, singular points, and Gabor filter reactions. Ridge-line flow is the global trace of the ridges, they give intuition about the orientation and curvature of the ridges. The orientation image is a representation of the local ridge flow. Most of the time, they are calculated block-wise on the fingerprint image and called the orientation map(OM). Singular points are, as mentioned before, the areas where a large change in the curvature occurs(the loop, in some research referred to as the core [90], [91], and delta), finally, the Gabor filter responses are obtained by applying Gabor Filters to the fingerprint image or orientation image. They are good at detecting ridges and valleys.

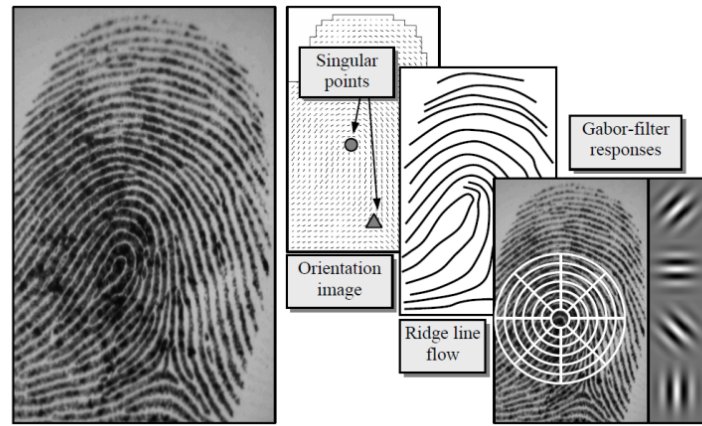


Figure 2.10 Features of Fingerprints that are used for classification [90]

A set of Gabor filters has been used on the image $I(x, y)$ in different frequencies with different orientations using the Gabor function $g(x, y)$ as defined by Equation 2.21 [92].

$$g(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \cos(2\pi \frac{x'}{l} + \varphi) \quad (2.21)$$

Additionally, a Convolutional neural network (CNN) was used as a trainable deep neural network. The fingerprint image of size 227×227 pixels was used to extract the CNN features in [92], which were used to train the CNN architecture.

- (a) Convolutional layers: In this step, the input images undergo convolution through the application of convolutional filters in the convolutional layer. Each image underwent a series of mathematical functions to generate a single value in the output feature map. The results were processed by an activation function to incorporate nonlinearities into the model. One of the most used activation functions used in their experiment was ReLU rectifier linear function, which they mathematically expressed as, $f(x) = \max(0, x)$, where x is the neuron input. An analytic function that provides a smooth approximation to the rectifier is available, $f(x) = \ln(1 + e^x)$, which is also called the soft plus function [92].
- (b) Pooling layers. The pooling layer is another basic part of the CNN architecture. This layer summarizes the features within a feature map extracted by the convolution layer using the average function or max-pooling, which was used to reduce the dimensions of the feature map. Therefore, the network had to learn to reduce the number of parameters and subsequently reduced the computational time. The Maxpooling algorithm extracted subregions of the feature map and keeps only their maximum value [92].
- (c) Fully connected layers. This layer classified the features extracted by the convolutional layers and down-sampled by the pooling layers. In this layer, every node was fully connected to all activations in the preceding layer. The CNN architecture contained layers, parameters, and functions are graphically illustrated in Figure 2.11 [92].

A pre-trained CNN network, NasNet-Large also has been used in many fingerprint researches. This model is a state-of-the-art CNN model that achieved the best classification accuracy on image-based datasets [93]. The other layers that followed the base layer are given in Table 2.1, and the training parameters are explained in Table 2.3 [93].

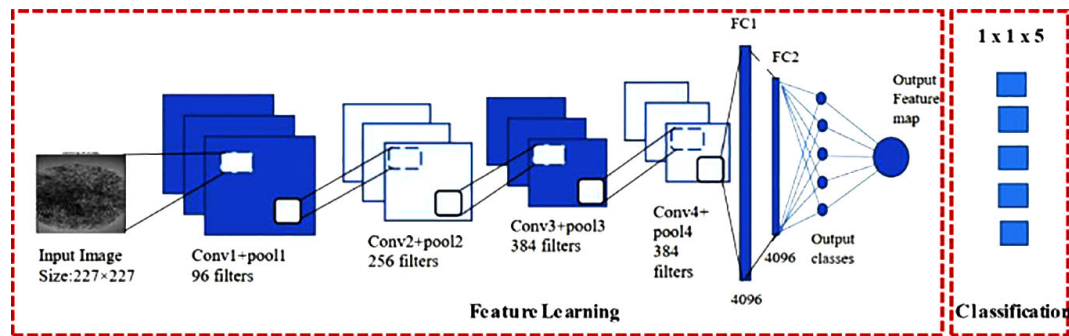


Figure 2.11 The graphical illustration of the CNN architecture [92]

Table 2.1 Layer Description of CNNs [93]

Layer (type)	OutputShape
NasNet-Large / Xception	(None, 331, 331, 3)
NasNet_function / Xception_function	(None, 8, 8, 2048)
GlobalAveragePooling2D	(None, 2048)
Dropout	(None, 2048)
Dense	(None, 4)

Table 2.2 shows the implication of error rates of different biometric systems.

Table 2.2 Implication of Error Rates[74]

Method	False Reject Rate	False Acceptance Rate
Fingerprint	3 to 7 in 100 (3-7%)	1 to 10 in 100,000 (.001-.01%)
Face Recognition	10 to 20 in 100 (10-20%)	100 to 1000 in 100,000 (.1-1%)
Voice Recognition	10 to 20 in 100 (10-20%)	2000 to 5000 in 100,000 (2-5%)
Iris	2 to 10 in 100 (2-10%)	$\geq .001\%$
Hand Geometry	1 to 2 in 100 (1-2%)	10 to 20 in 1000 (1-2%)
Signature	10 to 20 in 100 (10-20%)	2-5%

2.9 Related Works

2.9.1 History of Iris Biometrics

Several studies and research have been done and proposed in the field of iris biometrics identification systems which utilized identification techniques to identify people. This portion comprises an inspection of recent research, papers, thesis, and research that engaged traditional methods, machine learning, and deep learning methods in biometric and iris biometric identification systems. Deep learning algorithms, especially neural networks and convolutional neural networks (CNNs)

Table 2.3 Training Parameters of CNNs [93]

Parameter	Value
optimizer	Adam
loss	categorical_crossentropy
shuffle	True
Number of epochs	300
batch_size	64

have lately conducted advancements in diverse recognition systems and computer vision tasks such as object detection and recognition, image segmentation and captioning, and feature extraction.

The history of individual identification based on iris dates back over 100 years [5], [28], [29] [35], [41]. However, automatic iris identification is relatively more recent, dating back only decades ago [28], [29], [42]. In 1882, a French police officer Alphonse Bertillon carried out the first research trying to use eye measurements to identify criminals [5], [41], and later in 1886, Bertillon [5], suggested that the human eye can be used for biometric recognition [16]. Later, in 1936, Frank Burch [42] first proposed the use of iris patterns for human identification. In a clinical textbook “Physiology of the Eye” of 1953 which F. H. Adler [94] contributed a chapter, he described how the marking of the iris is so distinctive that the photographs of the iris can be used as a means of identification, instead of fingerprints. The perception of Birch was accepted by American ophthalmologists Aran Safir and Leonard 100 Flum that iris characteristic is individual for each person and different from another.

The first iris identification system to identify persons was introduced by Dougman, he introduced an influential system with high accuracy and speed, which has been applied to more than one database. The Gabor filter has been used to extract the iris characteristics, in the matching phase Hamming distance was used [95] By scrambling to simulate the structure and operation of the neurons in the human visual cortex over the usage of hierarchical multi-layer networks, deep learning has been revealed to be tremendously operative in automating the process of learning feature representation arrangements from the training data, thus eliminating the effortful feature engineering task. CNNs have their place in a specific category of deep learning methods designed

to process images and videos. By using repetitive blocks of neurons in the form of a convolution layer that is applied hierarchically across images, CNNs have not only been able to automatically learn image feature representations, but they have also outperformed many conventional hand-crafted feature techniques [96].

Comprehending behaviors to syndicate different physical biometric traits has underpinned many recent biometric identification system studies. Bouzouina and Hamami [97] offered a multimodal verification system that bonded the face and iris traits at feature-level assimilation. The research occupied several methods of feature extraction and applied support vector machines (SVM) algorithm for user confirmation and it produced an accuracy of 98.8%. Kien Nguyen, Clinton Fookes, Arun Ross, and Sridha Sridharan [98] proposed iris recognition with Off-the-Shelf CNN Features and explored the performance of state-of-the-art pre-trained CNNs on iris recognition. The research has shown that off-the-shelf CNN features, while originally trained for classifying generic objects, are likewise tremendously decent at representing iris images, effectively extracting discriminative visual features and achieving promising recognition results on two iris datasets: ND-CrossSensor-2013 and CASIA Iris-Thousand. Produced an accuracy of 98.7% at layer 6 on the LG2200 dataset and 98.8% at layer 5 on the CASIA-Iris Thousand dataset. With ResNet and Inception achieved similar peak recognition accuracies of 98.0% and 98.2% at layers 11 and 10, correspondingly, on the LG2200 dataset; and 98.5% and 98.3% at layers 12 and 10, respectively, on the CASIA-Iris-Thousand dataset. VGG, with its simple architecture, only achieved 92.7% and 93.1% recognition accuracy, both at layer 9, on the LG2200 and CASIA-Iris-Thousand datasets, respectively. Nada Alay and Heyam H. Al-Baity [99] proposed a multimodal biometric human identification system, which is based on a deep learning algorithm for recognizing humans using biometric modalities of the iris, face, and finger vein. The system is constructed using convolutional neural networks (CNNs) to extract features and classify images using a softmax classifier. Some deep learning researchers intended the use of finger veins for user recognition. For instance, Kim et al. [100] constructed a multimodal biometric recognition system based on CNN by fusing finger veins and finger shape. A ResNet pre-trained model has been used, and the two traits fused at the score level using some

fusion methods, which were weighted sum, weighted product, Bayesian rule, and perceptron rule. Lately, Boucherit et al. [101] employed finger vein traits to introduce an identification model, evolving a merge CNN model, several identical CNNs have been used with various input image qualities. Different experiments were operated using different network parameters and layers. The most optimal model acquired a mixture of original images and elevated images with the contrast-limited adaptive histogram (CLAH) method. The model was trained using the FV-USM, SDUMLA-HMT, and THU-FVFDT2 datasets, and it attained a recognition rate of 96.75%, 99.48%, and 99.56%, respectively. Furthermore, Liu et al. [99] have used the CNN algorithm in order to build a recognition system that obtained multiple finger vein images. They expanded the CNN model based on the AlexNet pre-trained model. The model has been trained using the SDUMLA dataset, and it obtained an accuracy rate of 99.53%. The global contactless biometrics market was measured at USD 18.69 billion in 2021 and is expected to grow at a CAGR of 15.2% during the forecast period [102], [103], as demonstrated in Figure 2.12.

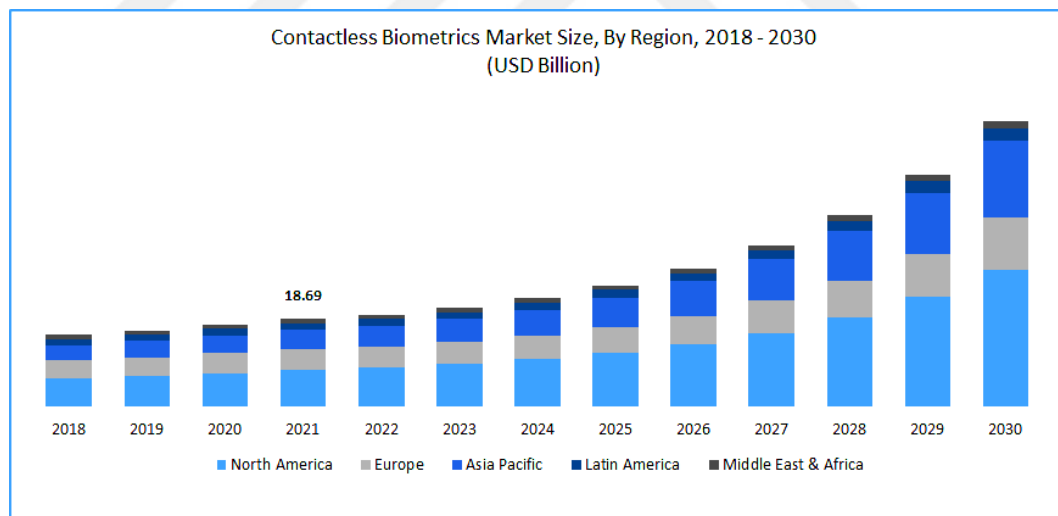


Figure 2.12 Contactless Biometrics Market Size [77]

In March 2022, Aware, Inc. applied adaptive authentication and announced the support of Flutter Version 2.10 into its Knomi mobile authentication platform [103]. HID Global and Temenos, a banking software company, joined forces in October 2020 to enhance user engagement for Temenos Marketplace customers. The collaboration integrates HID's multi-factor authentication solution into the platform. Temenos customers are spread across financial institutions and demand secure mobile and online transaction

experiences. HID Global is providing Temenos with its HID ActiveID authentication platform with risk profiling technology [102]. The Table 2.4 shows the Contactless Biometrics Market Report Scope.

Table 2.4 Contactless Biometrics Market Report Scope [77]

Report Attributes	Details
Market size value in 2021	USD 18.69 billion
Revenue forecast in 2030	USD 60.72 billion
CAGR	15.2% from 2022 - 2030
Base year	2021
Historical data	2018 - 2020
Forecast period	2022 - 2030
Quantitative units	Revenue in USD billion and CAGR from 2022 to 2030
Segments covered	By Component, By Authentication Mode, By Technology, By Industry Vertical, By Region
Regional scope	North America, Europe, Asia Pacific, Latin America; Middle East and Africa
Key companies	Aware Inc., BioConnect, Blue Biometrics, Cognitec Systems, Daon Inc., DERMALOG Identification Systems, Fingerprint Cards AB, Fujitsu Limited, HID Global, IDEMIA, M2SYS Technology, Neurotechnology, nVIAsoft, True Face, and Veridos GmbH.

2.9.2 Review of Selected Relevant Literatures

Much research has been done and is currently researchers are doing to get iris image identification to work well. Daugman [28], [51], proposed the traditional iris segmentation approach based on the approximation of the pupillary and limbic boundaries of the iris as circles. Given the acquired eye image, an integrodifferential operator was proposed for detecting the iris's inner and outer boundaries through an N-dimensional search of the image parameter space. Following the correct segmentation of the iris region, Daugman proposed a normalization process based on the rubber-sheet model which transforms the iris from a radial coordinate system to a polar coordinate system for size invariance. The normalized iris image is then convolved with 2-D Gabor filters in order to extract the rich textural information of the iris sample. He proposed Hamming distance (HD) as a measure of similarity between the iris template and other templates held in the database. Daugman's approach

yielded an impressive performance where high-quality iris image acquired under constrained imaging conditions is applied.

Correspondingly Wildes [52], proposed the iris localization boundaries by computing the binary edges followed by the Hough transform. Hough transform considers the set of edge points and finds a circle that best fits most edge points representing the iris boundary. Wildes also applied the Laplacian of Gaussian filters at multiple scales in order to compute the iris template. Correlation was adopted as a similarity measure in order to obtain the similarity between the iris templates. Wildes [52], represented the effectiveness of his approach. His approach shows an improved speed compared to Daugman's approach. However, these two traditional iris recognition systems are based on the assumption that iris boundaries are circular. The iris's pupillary and limbic boundaries are not perfectly circular or elliptical, which poses a significant challenge for shape-dependent models used in approaches like Daugman's and Wilde's. Also, in Daugman's approach, the search through the N-dimensional parameter space is inefficient and computationally intensive.

Alaa S. Al Waisy in [104] proposed a deep learning approach for iris identification biometrics, which incorporates two discriminative learning techniques: a CNN and a Softmax classifier. Their goal was to analyze the internal structures of the systems and determine their strengths and weaknesses. This would allow them to recommend an iris identification system that combines the best aspects of both techniques. The first one is CNN which is described it differs from traditional fully connected neural networks by combining a number of locally connected layers aimed at automated feature identification, followed by a number of fully connected layers aimed at classification. And the second one is the Softmax regression classifier, which was indicated as a generalized form of binary logistic regression classifier intended to handle multi-class classification tasks. In This work firstly, a preprocessing procedure was implemented based on employing an efficient and automatic iris localization to carefully detect the iris region from the background and all extraneous features, such as pupil, sclera, eyelids, eyelashes, and specular reflections. Their major reason for defining the iris area as the input to CNN instead of the whole eye image has talked to reduce the computational

complexity of the CNN. Another reason was to avoid the performance degradation of the matching and feature extraction processes resulting from the appearance of eyelids and eyelashes. The iris image normalization process was applied using Daugman's rubber sheet mapping [28], [51] in order to transform the iris image from Cartesian coordinates to polar coordinates. First, they conducted feature extraction and classification using a deep learning technique that combines a CNN and a Softmax classifier. To evaluate the accuracy of different configurations of CNN for a classification task, a cross-entropy loss function was utilized. This function measures the agreement between the predicted class scores and target labels and calculates the cost value accordingly..

In an endeavor to overcome these challenges, many new amendments and modifications or combinations of the traditional iris identification and iris segmentation approaches began to emerge in the literature. Liam et al. [21], proposed a simple method based on thresholding and function maximization in order to obtain the two ring parameters representing the iris's inner and outer boundaries. While in [22], Liu et al. adopted a modification of the circular Hough transform to improve the iris segmentation. To enhance the precision and stability of the Wildes method, they utilized a noise reduction technique and a hypothesis-and-verify approach. Also, they introduced an eyelid detection technique where the upper and lower eyelids are modeled as two straight lines. In another approach, Trucco and Raseto [23], proposed to find the optimal circle parameter of the iris boundaries by a simulated annealing algorithm. All these methods tried to either improve speed or accuracy but not both. Also, eyelids/eyelash occlusions as well as other noise factors were not given proper consideration. In [18], Masek suggested a revised approach that uses circular Hough transform to identify the iris and pupil boundaries. He then employed 1D log-Gabor filters for the extraction of the iris features while HD was used for template matching. Also, Tisse et al., [17] proposed a segmentation method based on the combination of integrodifferential operators and Hough Transform. His method was able to reduce computation time and excluded potential centers that lie outside of the eye image. However, it did not consider eyelash occlusions and noises which might influence the results of the algorithm.

Presently, researchers began to pay more attention to noise and occlusion detection in

iris segmentation [19], [34]. In [24], Haung et al., proposed a coarse segmentation of the iris using edge detection filters and Hough transform before normalization. The noise due to eyelids and eyelashes is then segmented by edge information based on the phase congruency method. In [25], Li Ma et al. proposed an iris identification approach based on characterizing key local intensity variations. The basic idea was to use the sharp local variation points to represent characteristics of the iris boundaries. Feature extraction was performed by constructing a set of 1D intensity signals to isolate the most important information from the original 2D image. A position sequence of local sharp variation points in such signals was captured as features, using a particular class of wavelets. A matching scheme based on the XOR operation was then adopted to measure the similarity between a pair of position sequences. These methods failed to consider low-quality iris images captured under less constrained imaging conditions which limits their applicability.

lately, the challenges posed by non-ideal iris images captured in less-constrained imaging conditions have started to catch the attention of researchers and scientists such as [10], [14], [20], [21]. In [105], Schuckers et al. tried two different approaches to handling “off-angle” irises. In both approaches, they sought to transform an off-angle image into an equivalent frontal image before segmentation. Also, He et al. [14], [20], presented an iris boundaries segmentation based on the pulling and pushing method. A cascaded Adaboost learning algorithm is first implemented to learn if the input image is an iris image and the rough position of the center and size of the iris in the image. The final segmentation of the iris boundaries is then achieved based on edge fitting and the pulling and pushing method. The eyelids were segmented using edge detection followed by a curve fitting process, while eyelash detection is achieved using a learned statistical prediction approach. Proenca and Alexandre [11], [21], proposed a segmentation method that first extracts the position and intensity feature information for each pixel of the iris image and applies fuzzy k-means clustering algorithm on the feature vector to produce a normalized intermediate image. The next step involves using edge map construction to identify the edges of the iris by analyzing the normalized intermediate image. Their results showed that this method significantly improved edge map-based segmentation

methods for non-ideal iris images.

Ryan et al. [106] presented an alternative fitting algorithm, called the Starburst method, for segmenting the iris. The iris image is first filtered using a smoothing filter and a gradient detection filter. Then, the darkest 5% of the image pixels was set to black, and all other pixels to white. A Chamfer image was then created where the darkest pixels in the Chamfer image are the pixel farthest from any white pixel in the threshold image. The starting point for iris segmentation was determined by using the darkest point of the Chamfer image. Next, they computed the gradient of the image along rays pointing radially away from the starting point. Two highest gradient locations assumed to be points on the pupillary and limbic boundaries were retained. The detected points were fit with several ellipses using randomly selected subsets of points. An average of the best ellipses was reported as the final boundary. The eyelids were detected using the active contour method. Pundlik et al. [107] presented another alternative segmentation algorithm based on graph cut. Their algorithm was a labeling routine instead of a fitting routine like the Starburst method. Their first goal was to assign a label - either “eyelash” or “non-eyelash” - to each pixel. Then after removing specular reflections, the gradient covariance matrix was used to measure the intensity variation of each pixel in different directions. Eigenvectors and eigenvalues obtained using the gradient covariance matrix are used to create a probability map, P that assigns the probability of having high texture in its neighborhood or not to each pixel. The “energy” corresponding to a particular labeling of the image pixel was written as a function of a smoothness term and a data term. The data term was based on a texture probability map. They treated the image as a graph where pixels were nodes and neighboring pixels were joined with edges, and they used a minimum graph cuts algorithm to find a labeling that minimized the energy function. The second objective was to categorize each pixel into one of four labels: eyelash, pupil, iris, or background. They used a method similar to the initial iris segmentation for this; however, this time they used an alpha-beta swap graph-cut algorithm. They used a geometric algorithm to approximate the iris boundaries with an ellipse and refined their labels accordingly. This is a computationally intensive method given the alpha-beta swap for all the pixels of the image is very tasking. Although most of these approaches yielded good results, many of them are based on supervised

learning which is computationally intensive.

Jus Lozej, Dejan Stepec, and Vitomir Struc in [108] proposed an iris identification system based on a deep learning approach. A segmentation step and a recognition step were used in their basic two-stage recognition pipeline. The pipeline was stated straightforward and very flexible, which replaced components and introduced manually annotated segmentation masks into the experimental setup. The input images were first fed into a segmentation model, with the purpose of extracting the mask corresponding to the iris area. This mask is then multiplied with the initial input image, masking out image regions. In their implementation they utilized a DeepLabV3+ [109] with a MobileNet [110] backbone and an output stride of 16 for segmentation and an Xception [111] model for recognition. They decided to use these particular models because these models give good results on a number of vision-related tasks and feature a relatively small number of parameters that need to be learned during training. Their final implementation had 29, 494, 217 open parameters, of which 2, 196, 033 came from the segmentation model, and 27, 298, 184 came from the recognition part.

Newly, the application of the active contour model for the segmentation of the iris boundaries has begun to attain the attention of many researchers [23], [24]. In [112], Daugman proposed a new method of iris image segmenting boundaries utilizing active contours based on the Fourier series approximations of the iris data. He adopted a statistical inference method for the detection and exclusion of eyelashes. Feature extraction and matching were conducted using Log-Gabor filtering and Hamming distance computation, respectively. Also, Shah and Ross in [23] proposed iris image segmentation boundaries using a geodesic active contour model. Firstly, the pupil is segmented by a binarization method followed through a circle fitting to the pupil's boundary. A contour is then initialized and evolved from the pupil boundary toward the iris boundary in order to determine the accurate boundary of the iris.

Caiyong Wang, Yuhao Zhu, Yunfan Liu, Ran He and Zhenan Sun in [113] proposed the Joint Iris Segmentation and Localization Using Deep Multi-Task Learning Framework. Their work contains an Encoder path and a Decoder path. To capture semantic information, the Encoder path encoded feature maps of CNN models by

convolution, ReLu, etc. The Decoder path decoded the feature maps to recover spatial information lost in the pooling layers by concatenation with feature maps of the Encoder path. The Encoder path adopted VGG-16 [114] as the encoding network. They removed the fully convolutional layers and the remaining network was used to learn hierarchical features. In their work the whole encoding network have been divided into 5 phases and every stage was composed of a series of convolutional layers, batch normalization layers, ReLU layers, and max-pooling layers. In lower stages, the feature maps contained more low-level spatial information such as edges but lack semantic information due to small receptive fields. In higher stages, bigger receptive fields extracted more semantic information and embed it in the feature maps. In fact, many similar networks, such as ResNet [67] and DenseNet [115], also have been used as the encoding network.

Zhao Zhe and Ajay Kumar [116] proposed a framework combining deep learning to study the iris domain problems and achieved superior experimental results than traditional methods. They have developed an optimized and unified deep learning architecture, referred to as UniNet, for both iris region masking and feature extraction, which is based on fully convolutional networks. The resolution after normalization was regularly set to 64×512 . Then applied a simple contrast enhancement process, which adjusted the image intensity so that 5% pixels were saturated at low and high intensities. The enhanced images then were used as input to the deep network for training and testing. For iris matching, they proposed FeatNet, and the input iris image is forwarded by several convolutional layers, activation layers, and pooling layers. They used FCN instead of CNN for iris feature extraction, and the reason for this selection was discussed that the most discriminative information of an iris probably comes from small and local patterns. FCN was stated to be able to maintain local pixel-to-pixel correspondence between input and output and therefore was a better candidate for the iris feature extraction.

Table 2.5 illustrates the literature review on Iris Segmentation Techniques for Iris Recognition System and the comparison between some segmentation techniques.

Table 2.6 shows the comparison between some pre-trained models such as IRISNet,

Table 2.5 Comparison table on literature survey[74]

NAME	METHOD	PERF	DA
High confidence visual recognition of persons by a test of statistical independence	Integrodifferential operator	High performance in iris recognition	Computational time is very high
Iris Recognition: An emerging biometric technology	Hough transform	Segmentation accuracy achieved up to an extent	Does not provide attention to EL as well as reflections etc
Recognition of Human Iris Patterns for Biometric Identification	Liber Masek's encoding algorithm	Localization of circular iris region as well as eyelids, eyelashes and also the reflections occurs	Speed of the system is low
Efficient and robust segmentation of noisy iris images for noncooperative iris recognition for Biometric Identification	Eight-neighbor connection based clustering	iris segmentation accuracy has been attained to an extent	segmentation of noisy iris images should be improved

AlexNet and VGGNet-16 .

Table 2.6 Comparison between pertained models[66]

Pertained CNN	Recognition Accuracy	Training Time
IRISNet	96.43%	01:15:30
AlexNet	95.09%	01:08:50
VGGNet-16	94.42	24:01:5

Table 2.7 explains the segmentation accuracy of some methods such as Daugman (integrodifferential operator), Wildes (Hough Transform), Proenca and Alexandre, Puhan, Modified integrodifferential operator, and Modified Hough Transform.

Table 2.7 Iris Segmentation Accuracy Result on different Methods[51]

Methods	Segmentation Accuracy Session 1	Segmentation Accuracy Session 2
Daugman(integrodifferential operator)	95.22%	88.23%
Wildes (Hough Transform)	98.68%	96.68%
Proenca and Alexandre	98.02%	97.88%
Puhan	98.49% (limbic) and 94.47% (pupil)	98.75% (limbic) and 87.17%
Modified integrodifferential operator	99.67%	99.35%
Modified Hough Transform	99.42%	98.53%

In this chapter, first, some key points about iris biometric backgrounds have been discussed, furthermore, iris image preprocessing includes Gaussian filter and Hough transform. Then iris image segmentation approaches are presented which contain Daugman's Integro-Differential Operator, Wildes' Hough Transform, and U-Net Iris Image Segmentation. Additionally, the background of iris image normalization with a homogeneous rubber sheet model has been reviewed. Moreover, a feature extraction technique based on the Histogram of Oriented Gradient (HOG), VGG – 16 and VGG – 19 Models, and ResNet Models were analyzed to highlight their strengths. Then Iris Pattern Matching based on the ORB algorithm was reviewed. After that, a short review of some related works that are significant to the proposed research is presented and discussed. on the contrary to all the research and developments in the area of iris identification systems and biometrics. Finally, Literature reviews were offered.

CHAPTER 3

Methodology

As it is designated in the title, this section includes the research methodology of this study and research. This section will cover a details explanation of the methodology that is being used to make this work and achieve a good result with high accuracy. The validity of discoveries and conclusions broadly depends on the quality of the research plan, information collection, information processing, algorithms, data examination, modules, and strategies that have been utilized. This chapter is devoted to the depiction of the strategies and methods done in order to get a high accuracy and appropriate performance speed within the iris identification system, how they will be executed, deciphered, and trained, and how the conclusion will be met. This phase is intended to explain how this research was created and will help to give it meaning and power as it will be true, straight, and analytical. All I mentioned will assist in the processing of iris identification and the methods of conclusions. It discusses the methodology and iris identification stages and the methods that were followed in the methodology and treatment. The analysis is discussed during deep learning and transfer learning, which have become favorite and trustworthy and have been used by many researchers. Deep Learning offers flexible and scalable techniques that don't need to be limited by the amount of feature extraction you need to do while training your algorithm.

3.1 Deep Learning and Convolutional Neural Network in Iris Identification

Iris identification models based on Convolutional Neural Networks (CNNs) have gained significant attention from researchers. This is because they overcome the limitations of conventional image processing methods, making the system more robust. [117], [118], [119]. By using repetitive blocks of neurons in the form of convolutional layers, pooling layers, and fully connected layers in Figure 3.1, CNN has not only the ability to acquire image feature representations but also outperformed many conventional hand-crafted feature techniques [119].

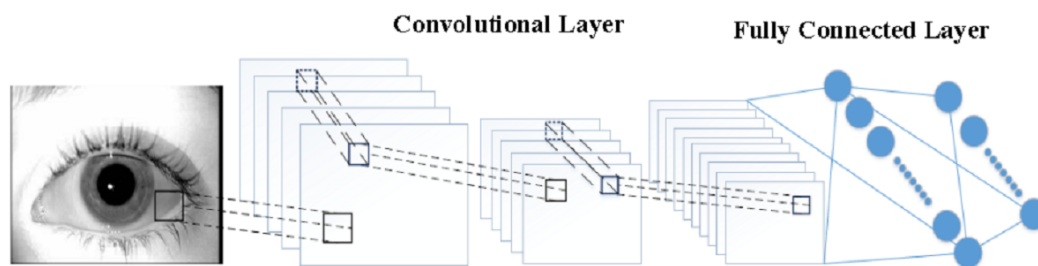


Figure 3.1 CNN architecture for Iris identification

Identification technology is concerned with recognizing the identity of individuals based on their unique physical or behavioral characteristics [120]. The physical characteristics such as iris, face, fingerprint, retina, vein, and hand geometry or the behavioral characteristics such as handwriting, human gait, signature, and keystrokes have unique, accurate, and stable information about a person to be used in authentication applications. The growing developments in information technology have made it possible to use biometrics in applications where it is required to establish or confirm the identity of individuals [120].

In recent times, technology has become an essential and foundational aspect of each person's life. Traditional methods are not efficient. Most of these methods are susceptible to loss, Hacking, forget or damage such as credit cards, identification cards, and even passports [121]. This has led to the need to work more secure and reliable methods of these traditional methods where a person can put documents and money and difficult to access by any person or Hacking. Biometrics can achieve this security for recognition and authentication, where each person has features and characteristics that couldn't own by two people, for instance, the lines found in a fingerprint, the sound signal (voice), the dimensions, and measurements of facial, the characteristics and feature of the iris, etc. [122]. Building a strong system with good specifications and ensuring speed and accuracy depends on the methods used to feature extraction, classification, and then recognition features. Deep learning can achieve these standards many researchers have begun using deep learning in their methods [123], and machine learning enables the system to automatically extract features, classify and detect patterns without explicitly programming them [124].

The structure of a perceptron is shown in Figure 3.2, where the inputs are multiplied by

the corresponding weights, which can be tuned to fit the model after the data, then added with a bias before being passed to an activation function. The activation function will pass the value to the output if the threshold is exceeded. Mathematically, the operation of a perceptron can be represented as shown below in Equation 3.1 [123]:

$$Y = \text{Activation}\left(\sum_{i=0}^{i-1} X_i \cdot W_{i,y} + B_{i,y}\right) \quad (3.1)$$

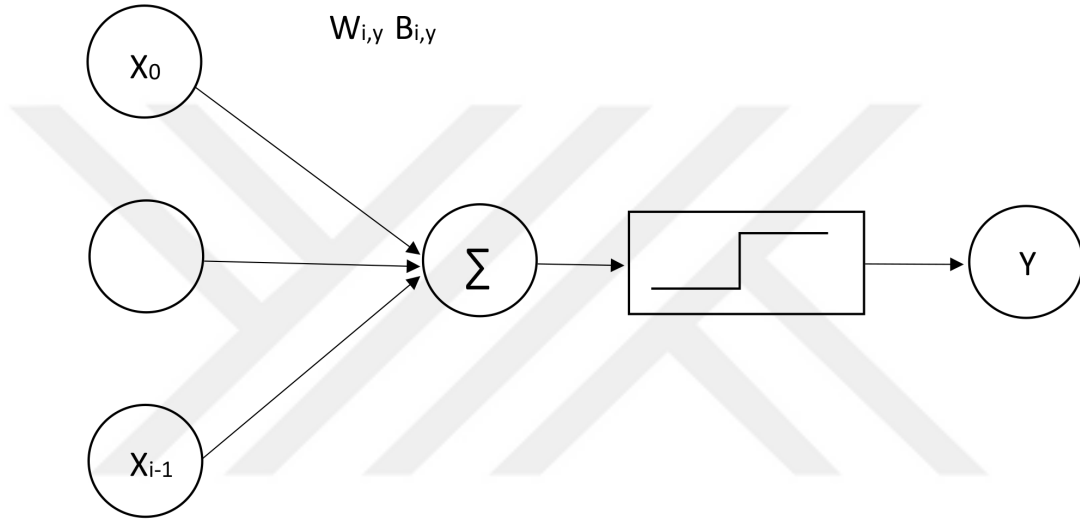


Figure 3.2 DL and CNN block diagram of a perceptron binary classifier

3.1.1 Convolution Layers

Convolution layers are the core process for extracting features from the input iris images. Randomly generated kernels/filters are used to perform the convolution operation and their values are updated by using the back-propagation algorithm. A standard convolution layer performs the following process in Equation 3.2 [117] [118].

$$X_j^L = f\left[\sum_{i \in m_j} x_j^{L-1} * w_{ij}^L + b_j^L\right] \quad (3.2)$$

Where X_j^L and X_i^L are i^{th} input from $(L-1)^{th}$ layer, and the j^{th} output to the L layer, respectively. The operation $*$ denotes the convolution operation and w_{ij}^L denotes the weights for the kernel b_j^L represents the bias term of the j^{th} output in the L layer $f(.)$ is

the activation function [117].

3.1.2 Pooling Layers

The main function of this layer is to down-size feature maps effectively for increasing the computation efficiency of the output of convolutional layers and intensify the useful features the filters show when convolving the image. The conventional pooling operation is fixed to be either a simple average, computed in Equation 3.3 [118]:

$$f_{avg}(x) = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.3)$$

or a maximum operation as Equation 3.4:

$$f_{max}(x) = \max x_i \quad (3.4)$$

where the vector x contains the activation values from a local pooling region of N pixels in an image. The architecture of the convolutional neural network and pooling layer is indicated in Figure 3.3 [120].

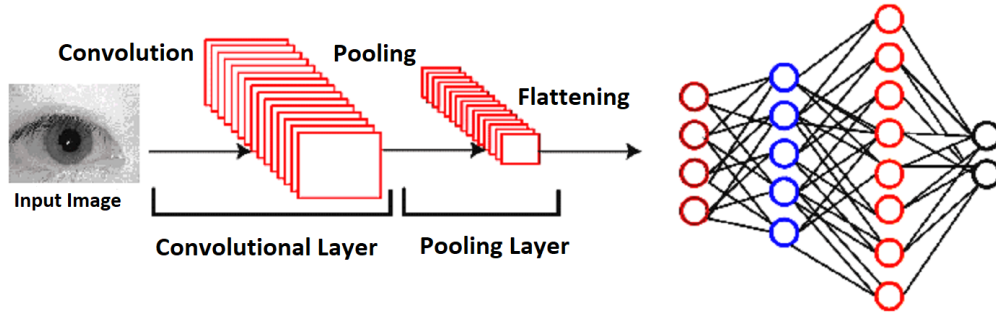


Figure 3.3 Convolutional Neural Network Architecture

3.1.3 Fully Connected Layers

The CNN pipeline is regularly terminated by fully connected layers. The Softmax function is an extension of the Sigmoid function. It compresses an N -dimensional vector, represented by x , to $\text{Sigmoid}(x)$. The output for each element is within the range of 0 to 1. The softmax function was used in diverse multi-class classification methods, especially in CNNs. In this case, the softmax layer is placed at the end of the

network and the dimension of the vector it operates on (i.e., N_c) represents the number of classes in the considered dataset. Additionally, the input of the softmax layer is the data generated by the last fully connected layer, and the output is the probability predicted for each class. The softmax function is defined in Equation 3.5 [118] [122]:

$$f_j(z) = \frac{e^{z_j}}{\sum_{j=1}^{N_c} e^{z_j}} \quad (3.5)$$

where z is the score from the fully connected layer.

3.2 Iris Identification Phases

This section discusses the iris identification stages, the proposed iris identification includes iris image acquisition, iris image preprocessing, iris image segmentation, iris image normalization, iris image feature extraction, and iris pattern matching. Figure 3.4 precisely describes the proposed iris identification stages.

3.2.1 Iris Image Acquisition

Iris image acquisition could be defined as the performance of obtaining an image from sources. The iris images are accumulated by capturing an image of the complete eye region of the human through some distinct digital camera devices and locating the pictures to the storage devices. The first step is the collection of iris images. The iris recognition system is a high-accuracy biometric identification technology with the privileges of uniqueness, constancy, and noninvasive, accurate matching performance. Iris image's modality and quality affect the performance of the detecting algorithms. The facility of usage and strength of the recognition is also affected by the image acquisition modality; therefore, iris image acquisition plays a significant role in the whole recognition system. Image acquisition is the operation of retrieving an image from a source, typically, a hardware-based source, in essence, it is merely to capture an image by using hardware.

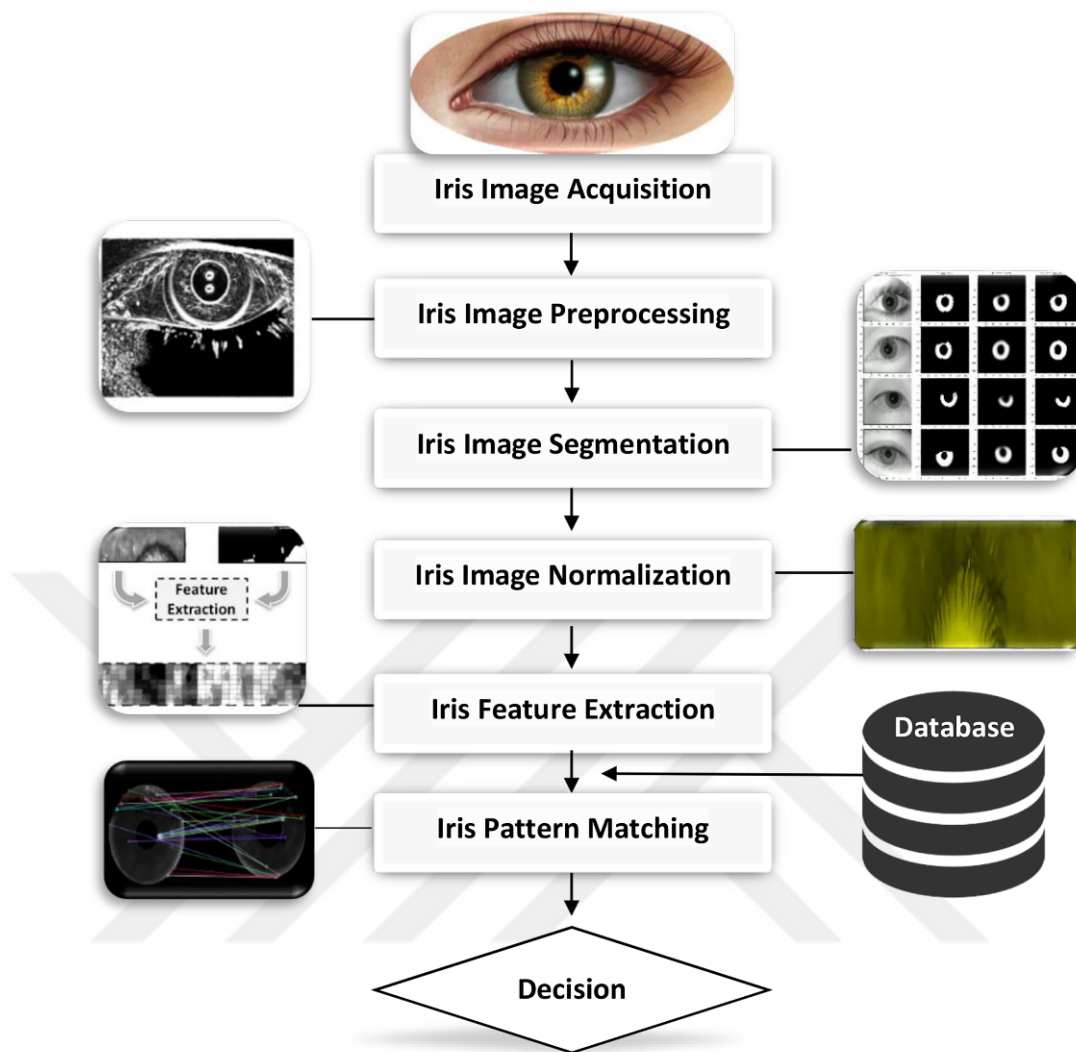


Figure 3.4 The proposed Iris Identification Flowchart and Architecture

In iris identification, the primary step is image acquisition. This step is very complex and complicated since the color and size of the iris of every person are unique. Due to the different lighting efficacy, the acquisition process might sometimes generate disparate results for the same person, aligning and different parting of distance. The prerequisite for iris image acquisition is the hardware design, the infrared light supply, lens, and CMOS sensor are significant features that affect the quality of iris image. As the salient point is stated hardware and image sensor selection acts quite an important role to acquire an ideal image. Image acquisition is a huge matter and crucial in the iris identification system, hence it is beyond my ability to do it by myself, though there are a large number of iris image databases.

3.2.1.1 Iris Databases

Currently, there are eight or even more permitted utilizable iris image datasets that are free to the public or academic use for iris identification biometric purposes, namely, they are the Chinese Academy of Sciences (CASIA [125]), with four discrete versions), University of Olomouc (UPOL [126]), University of Bath (BATH [127]), Multimedia University (MMU [128], two versions), Iris Challenge Evaluation (ICE versions of 2005 and 2006 [129]), West Virginia University (WVU MMU [130]), University of Beira Interior (UBIRIS MMU [131]), and Gaze Interaction for Everybody(GI4E [132]).

The key Purpose of the UBIRIS database which included UBIRIS.v1 and UBIRIS.v2 is to diminish the necessity of operator cooperation. UBIRIS.v1 version of the database is composed of 1877 images accumulated from 241 eyes throughout September 2004 in two separate sessions. It simulates less constrained imaging conditions. It is public and free available. UBIRIS.v2 version of the database contains over 11000 images, and continuously increasing, and more pragmatic noise factors [131]. Images were indeed captured at a distance and on the move. First, it should be strained that, with the exceptions of the UBIRIS databases and UPOL (imaged with an optometric device), all of the other iris databases contain NIR images [131]. Likewise, neither one of the data sets comprises images acquired at significantly varying distances and all of them used a rigid image acquisition protocol. Eventually, apart from the UBIRIS database, the residual databases contain the same moderate levels and kinds of noisy data [133]. GI4E - Gaze Interaction for Everybody, a public database of images was created for iris center and eye corner detection. The database contains a set of 1339 images acquired with a standard webcam, corresponding to 103 different subjects, and 12 images each [132]. The images are provided in PNG format which has a resolution of 800×600 pixels. Meanwhile, the images are correlated to a ground-truth text file. It contains manually annotated 2D iris and corner points in pixels [134].

3.2.2 Iris Image Preprocessing

Before iris image segmentation it is essential to perform image preprocessing in order to make the iris images present for segmentation. The objective of iris image preprocessing

is clearly to reduce and remove the noise from iris images, augment the data, and abridge the information and data in order to obtain an arbitrary iris image with a stable region and high quality. In addition, if a full face is given in the initial step, it is significant to detect the eye, which is extremely beneficial and significant in biometric systems, while Figure 3.6 exhibits eye detection from GI4E. To attain the purpose, the iris images should be filtered, and after that edge detection should be performed for iris image preprocessing [45]. Figure 3.5 demonstrates an eye image from the CASIA database and the results of the preprocessing stage performed.

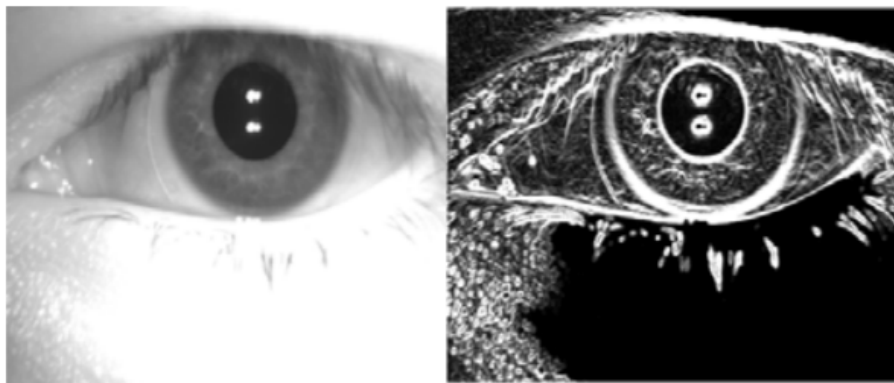


Figure 3.5 The results of the preprocessing stage performed

Face detection is performed using the Viola-Jones method [135]. The eye regions are localized based on anthropometric ratios as illustrated in Figure 3.6.



Figure 3.6 Eye and the Iris inner Detection

3.2.2.1 Iris Image Filtering

Iris image filtering is the initial phase in iris image preprocessing and after that iris edge detection will be performed, in order to omit the intrusion and impediment in the input iris image for edge detection. The important goal of an image preprocessing filter involves Noise Reduction, Color Normalization, Edge Detection, and Histogram Equalization, and iris image preprocessing filters are primarily used to support the high frequencies in the image or the low frequencies in the image.

Via efficient image filtering, Iris identification will be guaranteed. Before implementing the Mean Filter and Gaussian Filter, I am going to Implement an exponential operator. Given an exponent r compute for each input pixel P_{in} the output pixel value P_{out} with the following exponential operator formula as computed in Equation 3.6:

$$P_{out} = 255 \left(\frac{P_{in}}{255} \right)^r \quad (3.6)$$

Implement the function on the grayscale image which $r = 0.45$. Furthermore, Plot the histogram of the image before and after the transformation. Figure 3.7 illustrates the Histogram before and after the image exponential operator.

After the exponential operator, I implemented the histogram equalization operator on iris images from UBIRIS v2 database. The intensity of a pixel is specified $i \in [0..255]$, and the image histogram h , the total number of pixels of the image N , computes the cumulative distribution function in Equation 3.7:

$$PMF(i) = \frac{1}{N} \sum_{K=0}^i (h(k)) \quad (3.7)$$

And then use it to perform the equalization: $P_{out} = 255PMF(P_{in})$. Equalization distributes uniform pixel intensities amongst the assembled present range, which usually improves the contrast. Figure 3.8 illustrates the implementation of the function and plotting the histogram equation of the given image before and after the transformation.

In this research first, the mean filter and then the Gaussian filter have been applied, The Mean Filter is effective in blurring the image concerning to omit the noise. It implicates

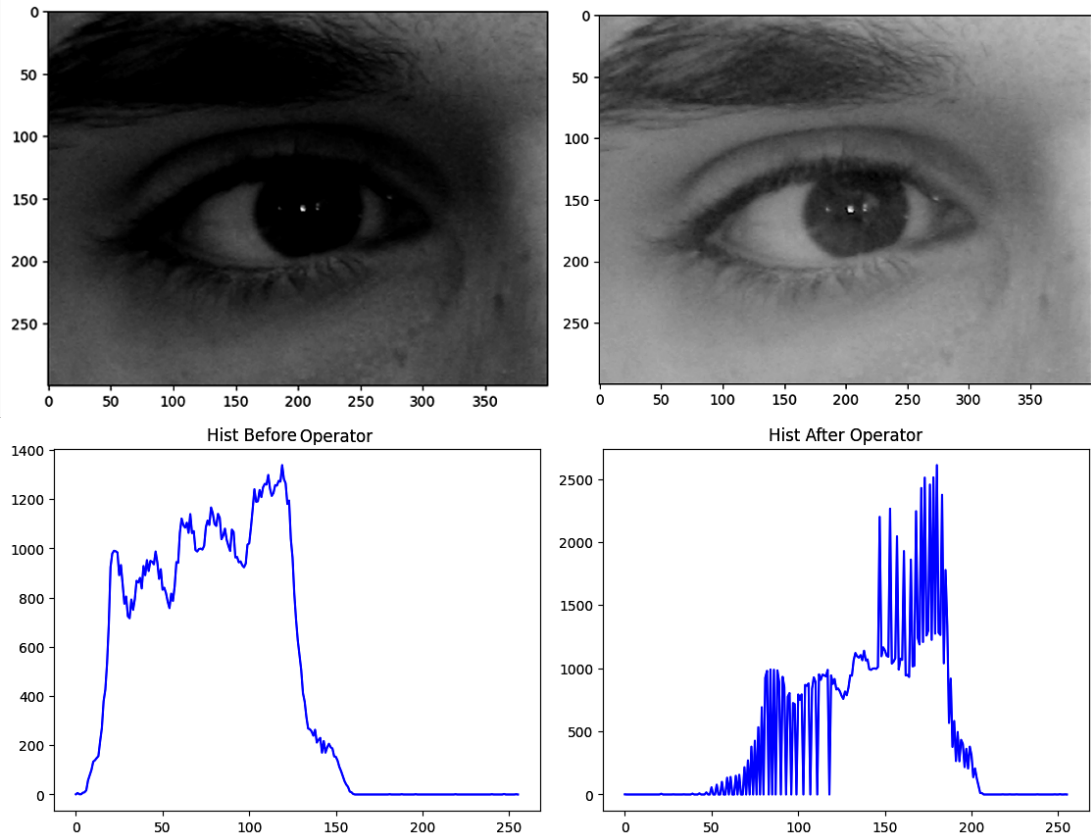


Figure 3.7 Histogram before and after image Exponential Operator

specifying the mean of the pixel values by $n \times n$ in this research I have applied a mean filter of kernel size 9×9 on an image. Proportionately the pixel intensity of the center element will be replaced by the mean. This eliminates some of the noise and smooths the edges of the image.

The Gaussian filter is then used to blur the required region of an image and dispel the noise with superior frequencies. Its performance is the same as the mean filter and is the linear filter, though illustrating the average weight uniformly, which blurs the edge effectively and reduces the noise. The Gaussian filter is a low-pass filter, whose 2D filter coefficients are computed as follows in Equation 3.8 [47]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.8)$$

Where x and y are the distances from the origin along the horizontal and vertical axes, and $\sigma = 0.8$ is the standard deviation of the Gaussian distribution. Hereby Figure 3.9 indicates the resultant of the Gaussian filter which $\sigma = 1.15$ to an image [47].

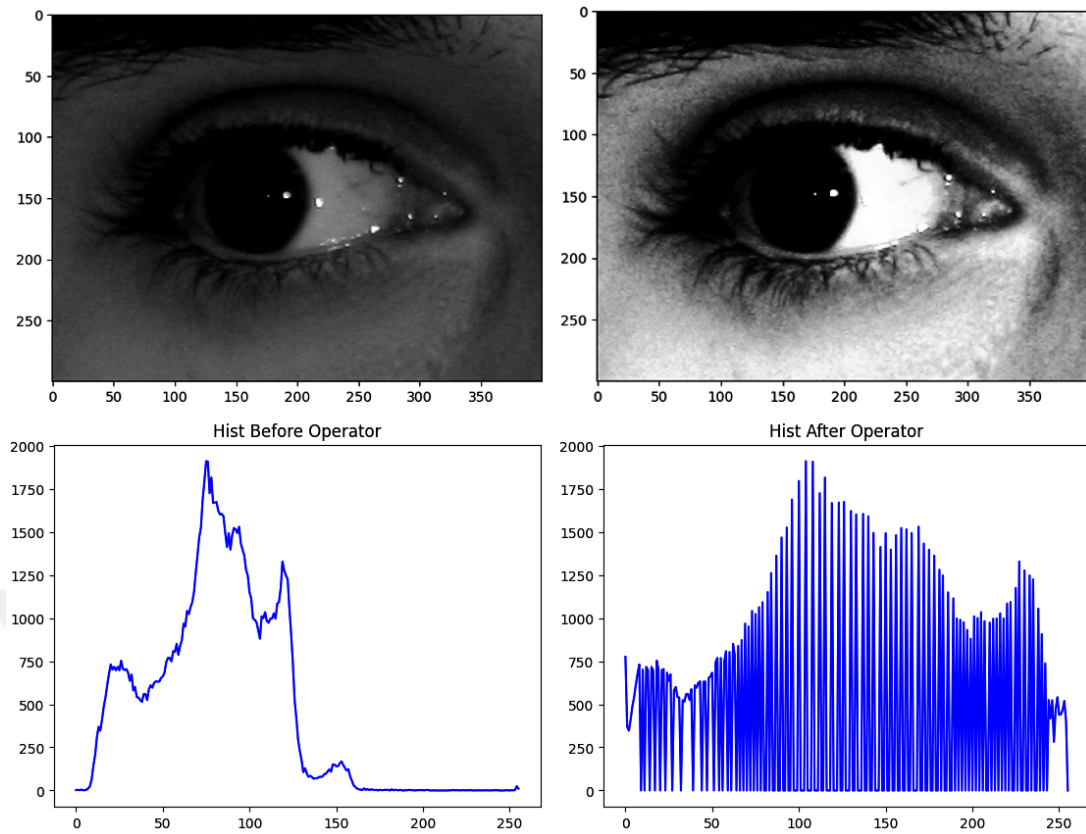


Figure 3.8 Histogram Equalization Implementation

3.2.2.2 Edge Detection

After filtering edge detection is an important part of the image processing method to find the borders of substances through the iris images, it detects interruptions in luminance. In areas such as image preprocessing, machine vision, and computer vision edge detection is commonly utilized to segment the images or data and extract features from the data. Most of the figures and shapes information of an image is surrounded by edges. First, of all using filters these edges are required to be detected in an image, and after that by boosting those parts of the image that contain edges, when the sharpness of an image enhances then the image will become more explicit [48]. Edge detection is one of the main parts of iris identification, The edge detection method forms pupil boundary detection precisely and easier. An edge typically happens on the boundary between two different areas in an image. To obtain an iris image, iris localization holds two edge detections, iris inner boundary detection, and outer boundary detection. An edge in an iris image is a significant local alteration in the image intensity. The pupil is a superior fragment of the eyes, by Hough transform could easily detect the

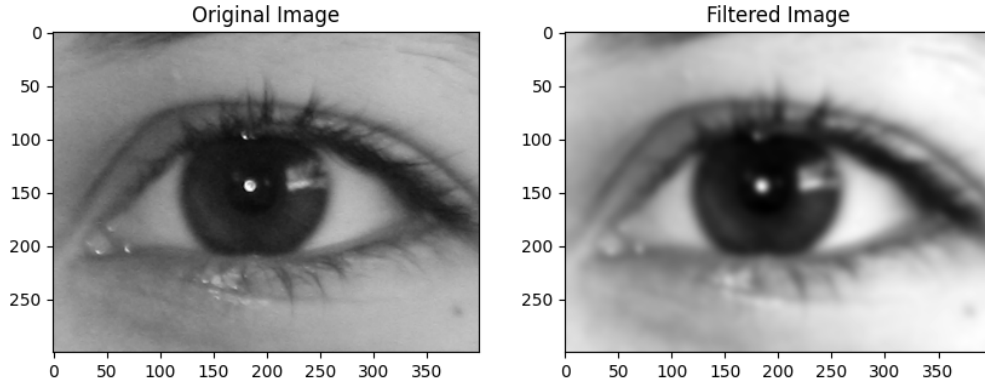


Figure 3.9 Gaussian filtering

boundary of the pupil by many methods such as Canny edge detection [48], Prewitt edge detection, Sobel edge detection, and the Otsu method can also be used to find the pupil [49]. which are proposed for this study. The measure of ellipticity and ellipse fitting are interesting to real-world image processing problems, Similar to other measures for preliminary geometric shapes. Ellipticity become famous in industry and nature, and the method of identifying completed it important in the identification system in the area of agricultural and medical imaging systems for identifying certain grains, onions, watermelons, cells, and even human faces as in this research, it is used for iris edge detection [136], as demonstrated the result of edge detection by ellipse fitting in Figure 3.10, and the equation is as follow in Equation 3.9:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \quad (3.9)$$

where (h,k) is the center of the ellipse, and $2a$ and $2b$ are the lengths of the axes of the ellipse. The longer axis is called the major axis, while the shorter axis is called the minor axis [136].

Next, the Canny edge detector has been performed, moreover, it is the first derivative of a Gaussian, and it approximates the operator that optimizes the product of SNR [50]. As shown in Figure 3.11, Canny edge detection can be appropriate in order to detect the iris. Canny edge detection is utilized to boost the iris outer boundary that is not identified competently in ordinary conditions.

Hough Transform is essential to be used in the case of detecting lines, circles, or any

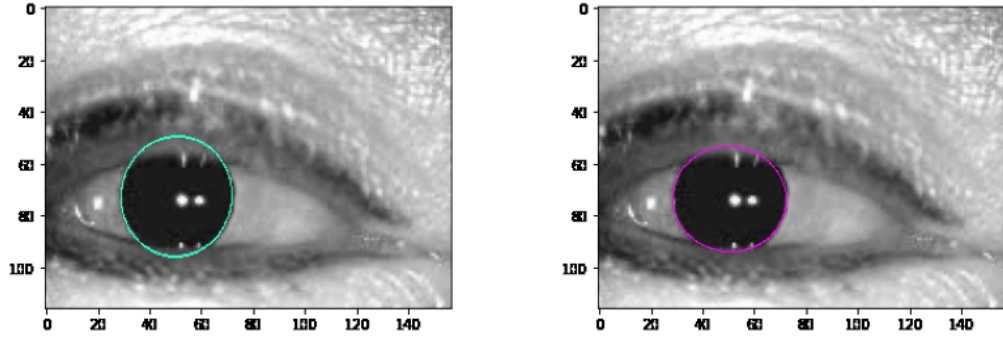


Figure 3.10 Ellipse Fitting Iris Edge Detection

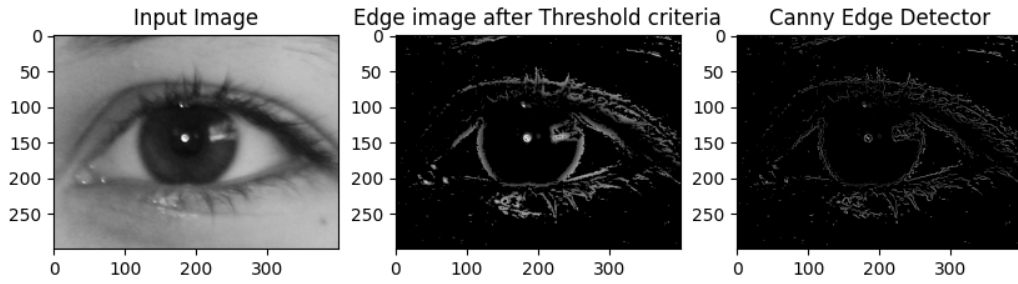


Figure 3.11 Iris edge detection by Canny edge detection

kind of parametric curves, first, it was introduced in 1962 by Paul Hough, and for the time it was used to find lines in images, especially it was an efficient method for detecting lines in binary images [137]. The aim is clear, to find the location of the lines in the images. Compared to the line the Circular Hough Transform in fact is simpler to each other, meanwhile the parameters of the circle can be directly transferred to the parameter space identical to the line Hough transform. The equation of a circle Hough transform is computed in Equation 3.10 [137]:

$$r^2 = (x - X_c)^2 + (y - Y_c)^2 \quad (3.10)$$

As we can see the circle obtains just three parameters, r , a , and b , where a and b indicate the center of the circle respectively in the x and y direction, and r illustrate the radius. So, we can get the parametric representation of the circle as follows in Equation 3.11 [137]:

$$\begin{aligned} x &= a + r \cos(\theta) \\ y &= b + r \sin(\theta) \end{aligned} \quad (3.11)$$

By increasing the number of parameters required to explain the figure and similarly the

dimensions of the spaceR parameter increase, likewise the complexity of the Hough transform. I have used Hough transform in order to detect the iris's outer and inner boundaries as the result is demonstrated in Figure 3.12.

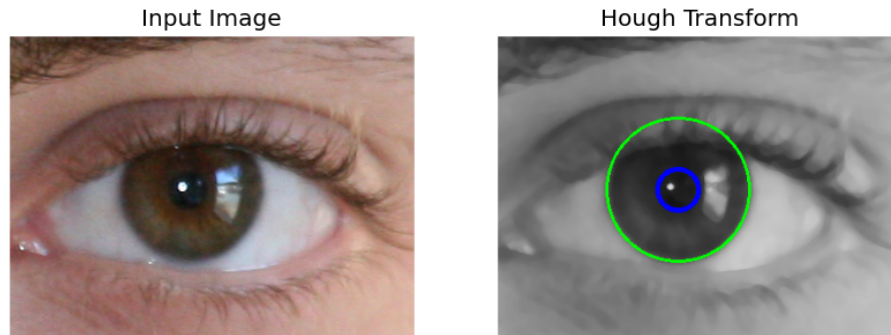


Figure 3.12 Hough Transform iris outer and inner boundary detection

3.2.3 Iris Image Segmentation

Iris image segmentation could be a significant section in the entire iris identification procedure and it is a very critical stage as well. If the iris local is not accurately segmented, the eyelashes, eyelids, reflection, and pupil noises would present in the normalized iris region. At the segmentation stage, the image is essential to be cautiously localized in order for the inner and outer boundaries of an iris can be modeled as a circle [21].

Through acquisition image of an eye does not include only an iris but it also contains pupil and data derived from the nearby eye region like sclera [51]. Thus, it is enormously significant to segment and localize the iris from the acquired eye image. Iris segmentation is the process of isolating the iris region from the rest of the acquired image. As explained earlier iris segmentation technology includes iris inner and outer boundaries localization, upper and lower eyelids detection and eyelashes, reflection, and pupil noises removal algorithms [21].

The key determination of the iris segmentation stage is to localize the two iris boundaries explicitly, the inner boundary of the iris pupil and the outer one of the iris sclera, and the last one is to localize eyelids [21]. Figure 3.13 shows a block diagram of the segmentation stage. As can be realized in Figure 3.14, the segmentation stage contains the following three steps:

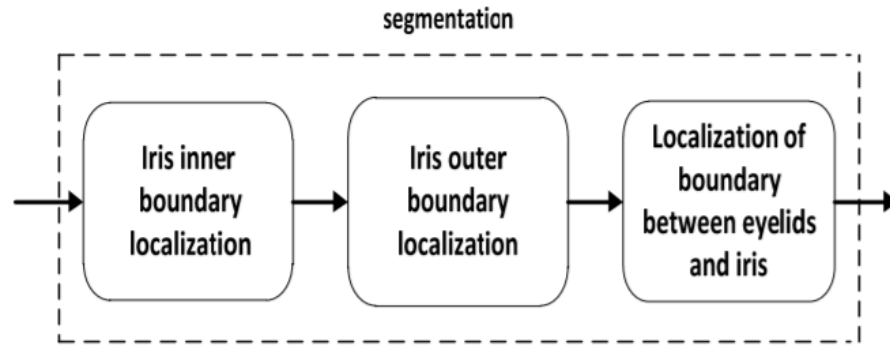


Figure 3.13 Block diagram of segmentation stage [21]

- Iris inner boundary localization (the boundary between pupil and iris) [21].
- Iris outer boundary localization (the limbic border between sclera and iris) [21].
- Localization of boundary between eyelids and iris [21].

Through the acquisition, the images of an eye do not include pure an iris, however, it also includes pupil and data apprehended from the around eye region like the sclera. Thus, it is enormously significant to segment and localize the iris from the acquired eye image. Iris segmentation is the process of isolating the iris region from the rest of the acquired image. In this study the proposed model for iris segmentation is U-Net.

In terms of segmentation, the U-Net model has illustrated good performance, especially semantic segmentation in biomedical image segmentation [53] such as Brain tumor segmentation in MRI images using U-Net [34], Liver and Liver Tumor Segmentation [54], Rethinking the U-Net architecture for multimodal biomedical image segmentation [55], Efficient Biomedical Image Segmentation on EdgeTPUs at Point of Care [56], and many more. The architecture of the U-Net model in this research with an iris image size of 240×320 pixels as an input has been used. The feature maps with (width x height) dimension along on the left represented by Individual box. Designed for Conv 1, two convolution operations by a 3×3 convolutional kernels were performed, with the size of 256×256 pixels the number of feature maps is 64, and alongside the max-pooling was performed with a kernel size of 2×2 . Figure 3.14 demonstrates the complete structure of the U-Net used in the iris

image segmentation, which stays a representative encoder-decoder architecture. The encoder part consists of a frequentative convolutional layer that has a kernel size of 3×3 , in addition, each pursues a 2×2 max-pooling by step 2, which regularly decreases the batch sizes. The decoder part contains the up-sampling processes with a 2×2 convolution, which splits the number of feature channels in half.

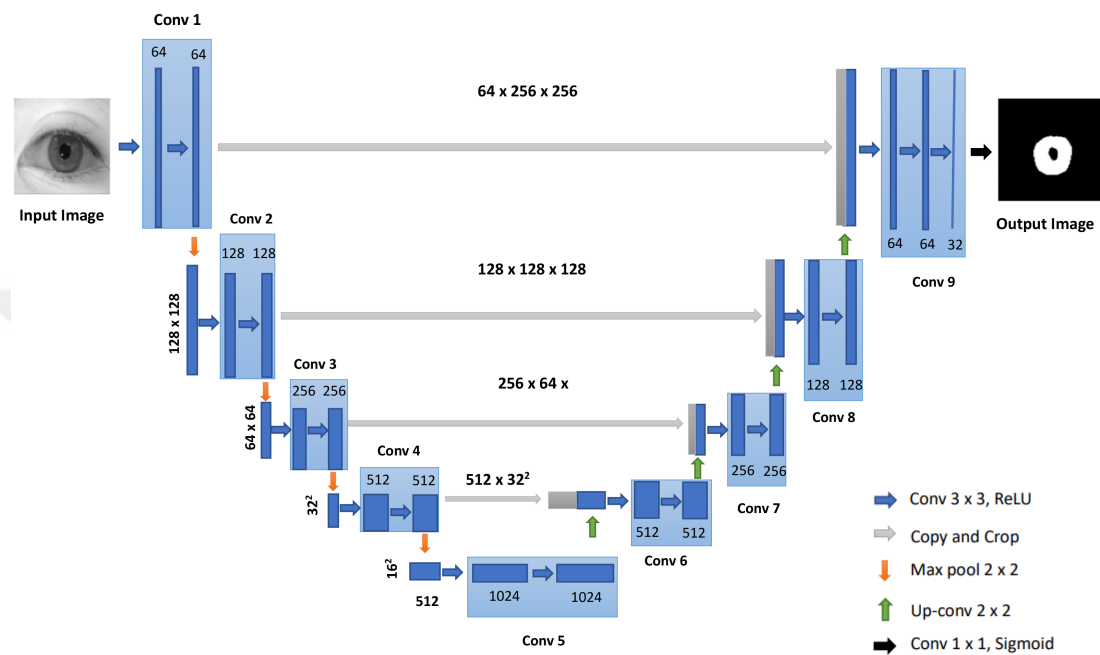


Figure 3.14 U-Net Flowchart and Structure used for Iris Image Segmentation

Figure 3.15: The Iris image segmentation outcome gained using the U-Net model. The figure exhibits from left to right, paradigm input images from the MMU iris dataset, the ground truth or mask images, the initial results generated by U-Net, and the finish the final result generated by U-Net. In the segmentation step, the iris region is transformed into a fixed-dimensional rectangular region, in order to make it easy to work with and extract features, iris image normalization must be performed. In post-processing the main threshold = 0.3, and on epoch function there has been used four more thresholds in order to predict the mask through the whole database. The threshold segmentation procedure can be considered as the process of separating foreground from background. Furthermore, it mainly extracts foreground based on gray value information, so it is especially useful for the segmentation of images with strong contrast between foreground objects and background [38]. In the Segmentation model training after 150 epochs attained 94.98% accuracy on the training set and 93.34% on

the test set. Four thresholds with the numbers 0.1, 0.2, 0.3, and 0.4 have been used.

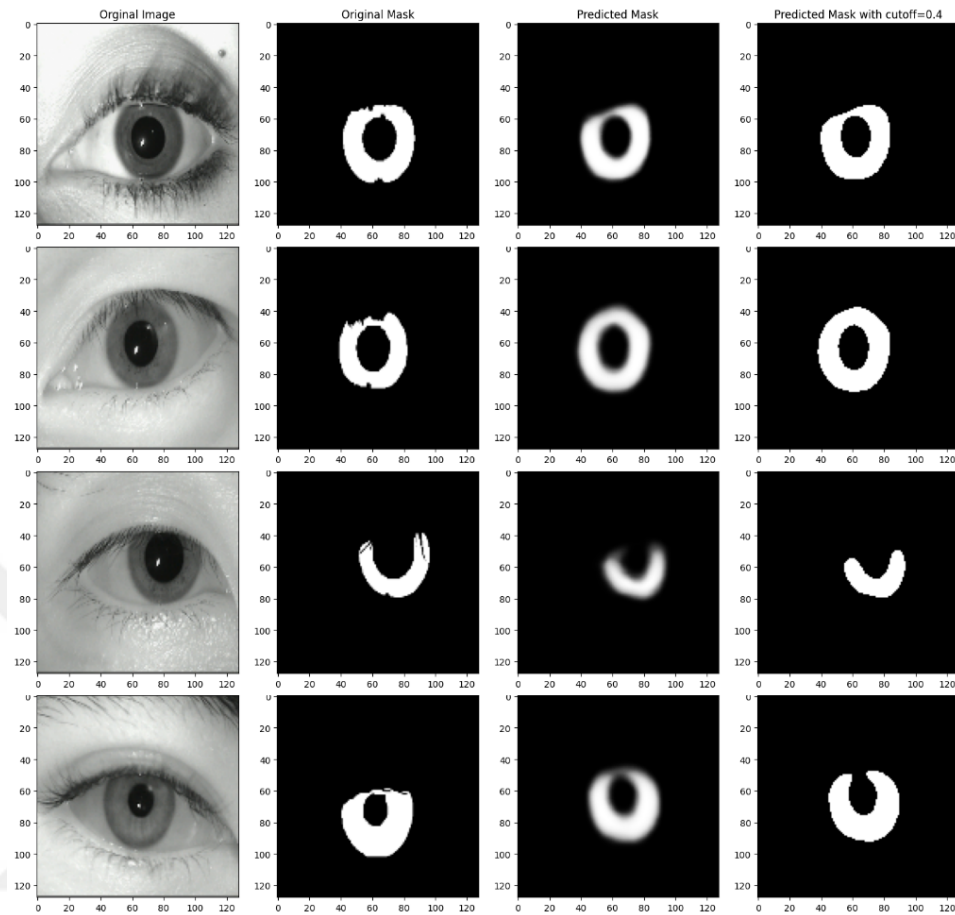


Figure 3.15 The result of Iris Segmentation using U-Net

In order to train the iris segmentation model, first I manually commented a tiny set of iris images and after that to learn the parameters of the CNN model using activation function (Relu and Sigmoid) and binary cross-entropy as training objective and adaptive moment estimation (Adam). The learning rate is ($1e-05$) with a batch size of 32 and with an epoch number of 100, and Table 3.1 shows the result.

Table 3.1 Segmentation Model Training Result (%)

Model Training Sets	ACC	EER	MSE
Training Set	92.25	0.1525	0.0457
Test Set	92.28	0.1521	0.0455

Furthermore, after training the model for iris segmentation with the CNN model I have used the U-Net pre-trained model of CNN with kernel_size of 3×3 , and the same as sample CNN using activation function (Relu and Sigmoid) and binary cross-entropy as training objective and adaptive moment estimation (Adam), and three thresholds (0.3,

0.4, and 0.5) have been used. The number of epochs is 150 and batch_size is 32, and Table 3.2 demonstrates the result.

Table 3.2 Segmentation Model Training Result (%)

Model Training Sets	ACC	EER	MSE
Training Set	94.98	0.0980	0.050
Test Set	93.34	0.1395	0.0506

3.2.4 Iris Image Normalization

Iris images might be taken in several sizes and with different distances, as a result of brightness differences, the circular and radial size of the pupil may vary appropriately. Thus, the iris area has to be normalized to reimburse for this alteration. When the iris identification technology acquires an image from a person's eye. Afterward, the image of this iris needed to be segmented and then normalized for the feature extraction operation. The proficiency and implementation of iris identification systems particularly depend on segmentation and normalization. The normalization process normalizes iris images to a constant dimension and sizeable fragment to avoid its incompatibility that might arise for various reasons such as iris imaging remoteness discrepancy, a revolution of the device, illumination variation, iris stretching, eye rotation, head deformation and etc. Consequently, iris image normalization is applied to decrease all of these obstacles. Therefore, in the iris normalization section. The Hough transform algorithm for detecting the iris's inner and outer boundaries has been implemented as discussed previously in image preprocessing with the equation as well, and then Daugman Normalization has been applied. This model is Invented by John Daugman [29], [49]. The Homogeneous Rubber Sheet Model which was devised by Daugman and proposed usage of a Rubber-Sheet Model, assigns each point in the iris region as pair antithetical coordinates (r, θ) where r is the distance or the range between $[0, 1]$ while θ denotes the angle $[0, 2\pi]$ [58]. The model is illustrated in Figure 3.16.

To reallocate the iris area, the formula from (x, y) in Cartesian coordinates to normalize nonconcentric and abnormal polar is defined as follows in Equation 3.12, Equation 3.13 and Equation 3.14 [29]:

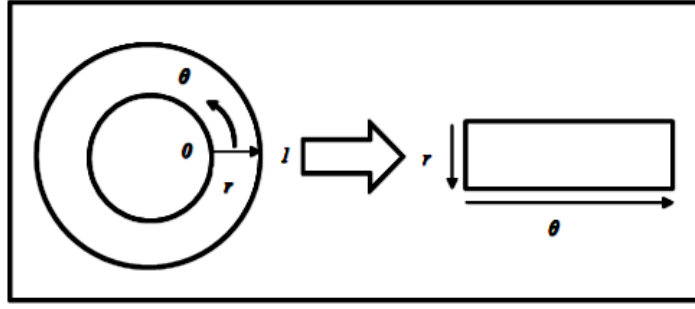


Figure 3.16 Rubber sheet model [29]

$$I[x(r, \theta), y(r, \theta) \rightarrow (r, \theta)] \quad (3.12)$$

$$x(r, \theta) = (1 - r)x_p(\theta) + rx_1(\theta) \quad (3.13)$$

$$y(r, \theta) = (1 - r)y_p(\theta) + ry_1(\theta) \quad (3.14)$$

where $I(x, y)$ is represented as the iris region (x, y) indicates the original Cartesian coordinates, (r, θ) demonstrates the corresponding normalized polar coordinates, x_p, y_p and x_1, y_1 are the coordinates of the pupil and iris boundaries along θ direction [29]. The iris within the perceived image is detected by discovering the two closely concentric circular edges, explicitly the two boundaries among the pupil, the iris and the sclera. Figure 3.17 demonstrates the normalized iris image in the proposed research. The left image illustrates the observed image of the iris's inner and outer boundaries detection, in the middle is the segmented iris, and the right image represents the normalized iris image.

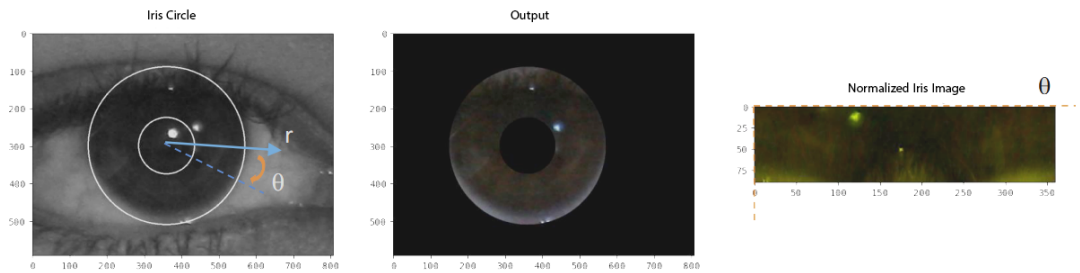


Figure 3.17 The Normalized iris image

Normalizing the input data of neural networks to zero-mean and constant standard

deviation has been known for decades [138] to be valuable to neural network training. With the intensification of deep convolutional neural networks, Batch Normalization (BN) naturally extends this idea across the intermediate layers within a deep convolutional neural network [139]. Nowadays, there is slight disagreement in the deep learning community that BN accelerates training, enables higher learning rates, and improves generalization accuracy [139] and Batch Normalization (BN) has successfully proliferated throughout all areas of deep learning [140], [141]. Batch Normalization similarly has a beneficial effect on the gradient flow through the network, by decreasing the dependence of gradients on the scale of the parameters or of their initial values. This allows for much higher learning rates without the risk of divergence [142]. Furthermore, batch normalization regularizes the model and decreases the prerequisite for Dropout. The batch normalization equations are computed in Equation 3.15, Equation 3.16, Equation 3.17 and Equation 3.18 [140]:

$$\mu\beta = \frac{1}{m} \sum_{i=1}^m x_i \quad (3.15)$$

$$\sigma_{\beta}^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu\beta)^2 \quad (3.16)$$

$$\hat{x} = \frac{x_i - \mu\beta}{\sqrt{\sigma_{\beta}^2 + \epsilon}} \quad (3.17)$$

$$y_i = \gamma\hat{x}_i + \beta = BN_{\gamma,\beta}(x_i) \quad (3.18)$$

The *BN* transform was added to a network in order to manipulate any activation. In the notation $y_i = BN_{\gamma,\beta}(x_i)$ indicates that the parameters γ and β are to be learned [140].

The normalization consists of transforming the region of the iris, which is represented in the form of rings, to a rectangular shape with fixed size [143], [144]. This condition requires a transformation of Cartesian coordinates to polar coordinates. Recall that each point making up the region of the iris is characterized by its coordinates (x, y) in Cartesian theory and its intensity of gray level $0 \leq \text{gray } x, y \leq 255$. Passing to polar coordinates, they obtained the following system, computed in Equation 3.19 [144]:

$$\begin{aligned}
0 < \theta &\leq 2\pi \\
P_{pupil} &\leq P \leq P_{iris} \\
Gray(P, \theta) &= gray(x, y)
\end{aligned} \tag{3.19}$$

where P_{pupil} is the radius of the pupil, and P_{iris} is the radius of the iris. they obtain a frame “raster” whose lines are marked by P and columns by θ [144].

3.2.5 Iris Image Feature Extraction

When the iris images have been magnificently segmented and normalized, the afterward phase is to extract considerable data from the iris pattern that has been obtained from normalized iris images. The features which are extracted from iris images are programmed to produce iris patterns. Several iris identifications take the benefit of using bandpass decomposition of iris image to produce biometric template [60]. In this research work first of all I have used a Histogram of Oriented Gradient (HOG). HOG is one of the well-accredited and suitable features for feature extraction as a consequence of its preferable performance and fairly simple computation. Since first proposed in [62]. HOG due to its good accuracy and efficiency has been broadly used in numerous image processing and object detection algorithms [145]. Figure 3.18 shows the Gradient extraction from the iris image, the image gradient measures the change of intensity [146]. Mathematically, for a two-dimensional function $f(x, y)$ the gradients can be computed by the derivatives with respect to x and y [146].

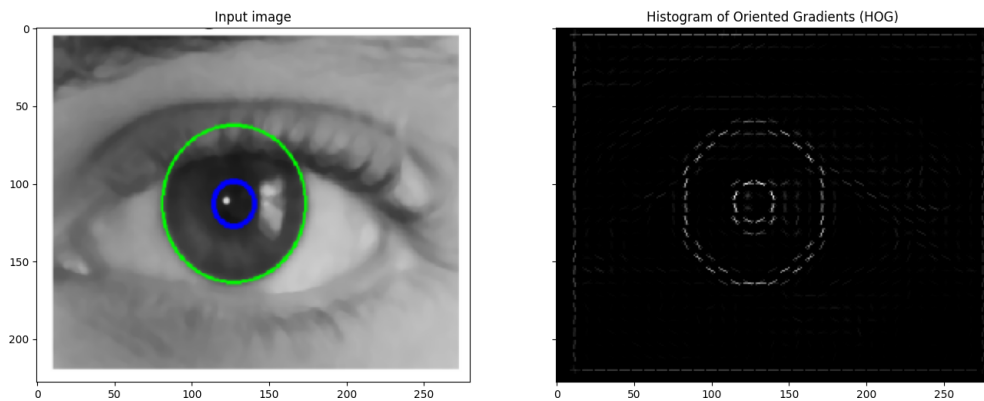


Figure 3.18 HOG Iris Feature Extraction

Meant for a digital image where x and y are distinct values, the derivatives can be approached by limited differences [147], and the gradient location x,y can be computed in Equation 3.20 and Equation 3.21:

$$G_x = I(x+1,y) - I(x-1,y) \quad (3.20)$$

$$G_y = I(x,y+1) - I(x,y-1) \quad (3.21)$$

Here G_x and G_y characterize the gradients in the horizontal and vertical directions individually [147]. Figure 3.19 shows the HOG iris feature extraction result in this research.

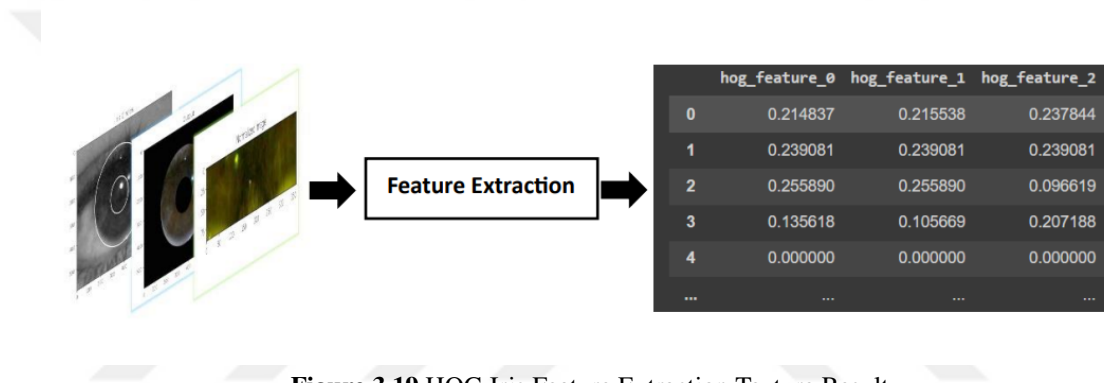


Figure 3.19 HOG Iris Feature Extraction Texture Result

I have constructed a group of CNN methods from scratch using transfer learning and these all methods pretend to be learning, though, there might be still abundant work for supplementation. To improve the models, I have spent a long time trying different configurations, altering the learning rate, adding more layers, regulating the neuron numbers per layer, and much more work has been done like re-modeling and evaluating the models. Meanwhile, performing all this is wasting time. Pleasingly, there is a branch of other methods and knowledge I used to prevent wasting time and that is called transfer learning, a model can learn from another problem, in order to use them for my own problem. In addition, these models are pre-trained like VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet, and InceptionV3 have been performed with the MMU Iris dataset and UBIRIS v2, the reason these models are used since they are fully convolutional. The extraction of features is an essential factor of the iris popularity framework. Previously, a variety of features were used to track the iris popularity framework, the claim of the sufficiency acquired through the practice

of convolutional neural networks (CNNs) to iris identification and recognition has fascinated considerable interest due to the achievement of a high level of expertise and high accuracy in iris identification [148].

As mentioned perversely, in this research, I have used several models to extract iris features, additionally, I peruse the susceptibility and capabilities of a convolutional neural network comprehended using the VGG16 method, mostly known as a convolutional community model. Figure 3.20 represents the VGG-16 Iris Feature Extraction Architecture.

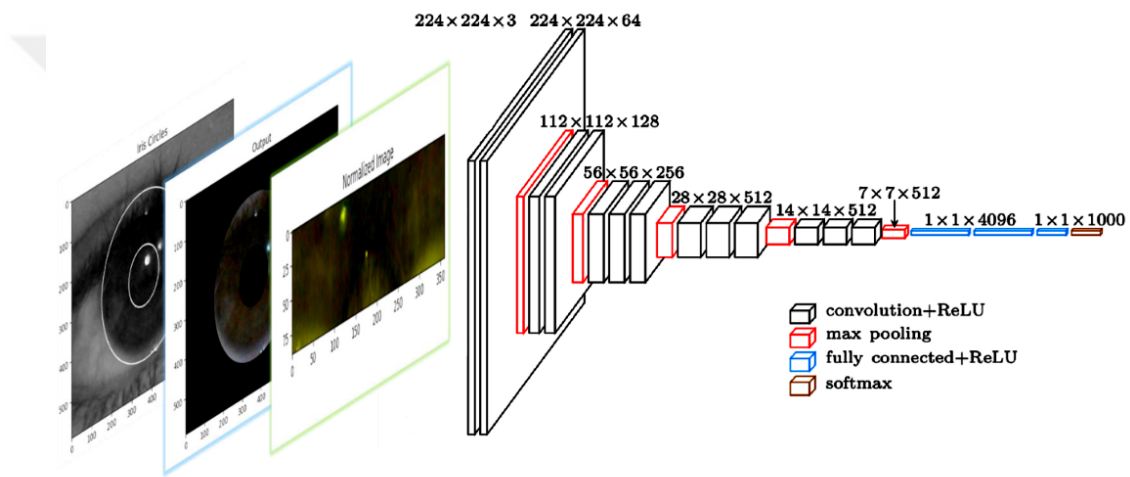


Figure 3.20 VGG-16 Iris Feature Extraction Architecture

The VGG-19 model is also proposed to extract the features of the iris images in this research, as it is the modified version. The concept of the VGG-19 model (VGGNet-19) is similar to the VGG-16 except that VGG-19 contains 19 layers to be performed. In the convolutional layers, the numbers 16 and 19 indicate the number of weight layers. Respectively the difference is between the layers, VGG-19 has three extra convolutional layers compared to the VGG-16. VGG-19 is a CNN architecture that consists of 19 layers of deep neural network in addition it possesses more weight. VGG-19 in input takes an image size of 224×224 , VGG's convolutional layers influence a minor amendable scope of 3×3 content to increase the depth, the minimum possible size, that is pursued by a ReLU activation function. The VGG19 contains three fully connected layers [149]. Figure 3.21 demonstrates the VGG-19 Iris Feature Extraction Architecture.

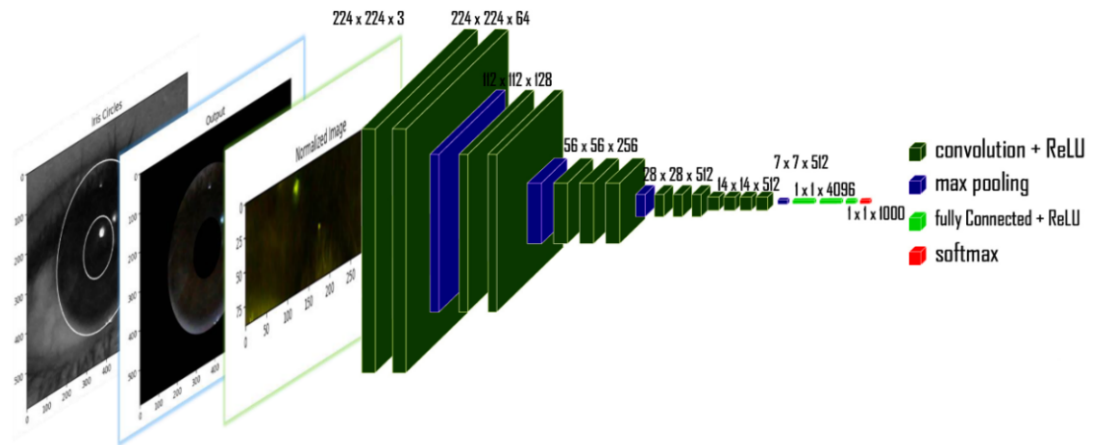


Figure 3.21 VGG-19 Iris Feature Extraction Architecture

Basically, the ResNet model architecture is taken from ResNet-34 and includes 34 weighted-layers, that support different ways to enhance further convolutional layers to a CNN. The established network originally was founded on the VGG neural networks (VGG-16 and VGG-19), as mentioned previously each convolutional network has a 3×3 filter. Though, the ResNet pre-trained model contains minor filters and is less complicated than a VGG-Net. ResNet-50 is fully convolutional and contains 50 layers deep, the 50 layers in ResNet use a bottleneck meant for the building block, which decreases the number of parameters and matrix multiplications, this characteristic empowers greatly faster training of each layer [150]. ResNet-101 is the same as ResNet-50, a fully convolutional neural network but contains 101 layers deep. Figure 3.22 describes the Distribution of pre-trained CNN Models.

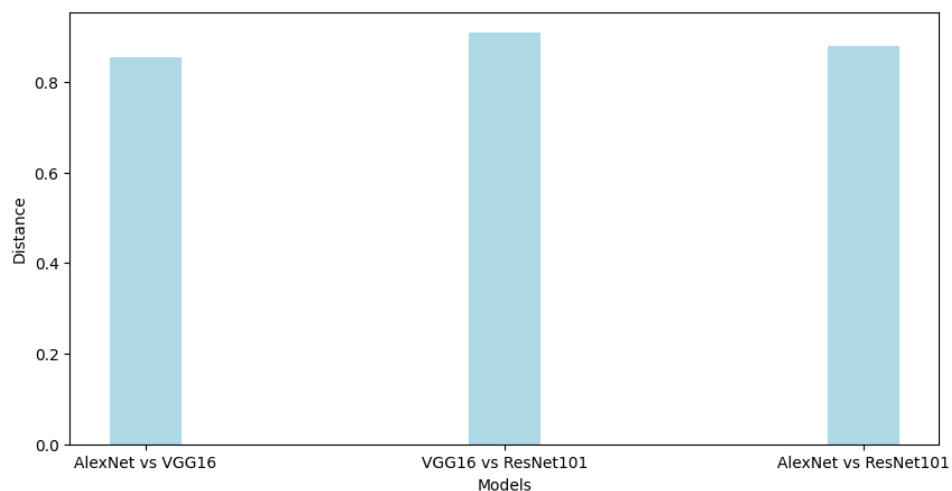


Figure 3.22 Distribution of pre-trained CNN Models

3.2.6 Iris Image Pattern Matching

The process of calculating the rating of resemblance amongst the input iris images such as test set and training set from the database and purposes to extract the most perceptive features used to produce the significant iris mystery. In the matching stage, the patterns are generated at the iris feature extraction stage to measure the similarity between two images of iris patterns. This stage measures the similarity and dissimilarity between the two binary codes for making a decision of acceptance or rejection [61]. In this research, for iris pattern matching, the ORB algorithm is utilized by me, as the matching technique and method. on this basis, the enhanced ORB algorithm is suggested. Image matching is used to find the similarities between the original mask and the mask which is predicted in the segmentation section, algorithm in order to detect feature points. The ORB image matching algorithm is normally divided into three stages: feature point extraction, generating feature point descriptors and feature point matching [69]. The flow chart is exposed in Figure 3.23.

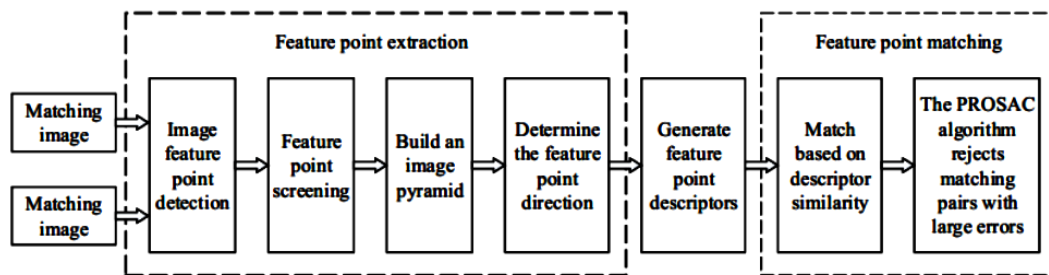


Figure 3.23 Image matching flow chart based on ORB algorithm [69]

- (a) Feature point extraction: The ORB algorithm uses the improved FAST (features from accelerated segment test) [69] algorithms in order to detect feature points.
- (b) Generate feature point descriptors: Then after extracting the Oriented FAST feature points, the ORB algorithm applies the enhanced BRIEF algorithm [70] to calculate the descriptors for separate points. BRIEF is a binary vector descriptor, in which the vector contains a number of 0 and 1 [71].

Where $p(x)$ can be the gray value at the ground x about the image feature point, and $p(y)$ stays the gray value at the ground y about the image feature point and mathematically computed in Equation 3.22. [71].

$$\tau(p_{ix,y}) = \begin{cases} 1, & p(x) \leq p(y) \\ 0, & p(x) > p(y) \end{cases} \quad (3.22)$$

- (c) Feature point matching: After the first and the second steps it is determining the scale and rotation statistics of the image feature points, as it is essential to specify the similarity between the feature point descriptors in the two different time images to determine whether they match or not [71] [72].

In this research first I applied the ORB algorithm to find the similarity between the ground truth mask and the predicted mask from iris segmentation, and later on, the two iris images were applied to match the similarity. Figure 3.24 describes the matched key points of images and Figure 3.25 demonstrates the matched points which were applied on the same iris image, ground truth mask, and the predicted mask from iris segmentation. ORB uses BRIEF descriptors however the BRIEF performs poorly with rotation. So what ORB does is rotate the BRIEF according to the orientation of core points [71]. Using the orientation of the patch, its rotation matrix is found and rotates the BRIEF to get the rotated version.

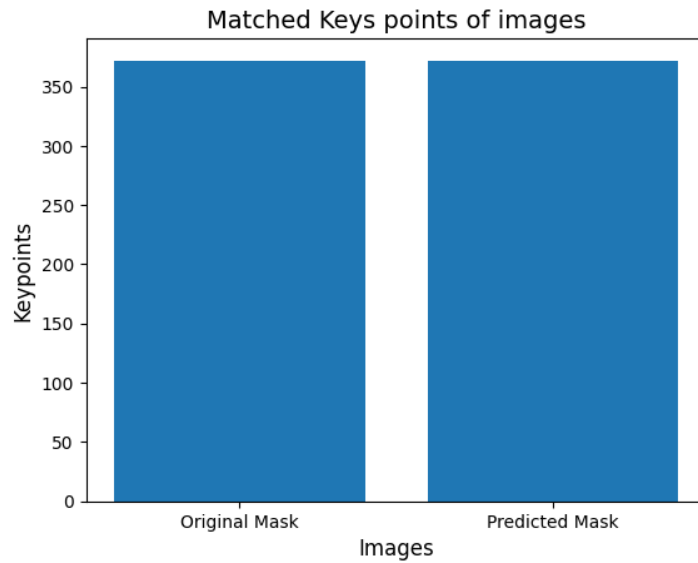


Figure 3.24 Matched Keys points of the same iris

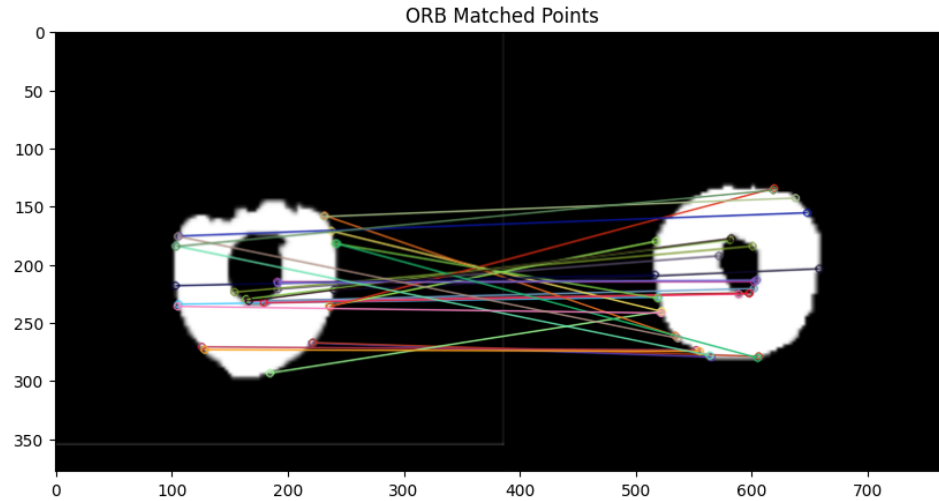


Figure 3.25 ORB Matched Points on the same iris

Figure 3.26 illustrates the matched key points of two different images from different persons and Figure 3.27 represents the matched points which were performed on different iris images. In the pattern matching section, it was found that the matching points of the two images are different.

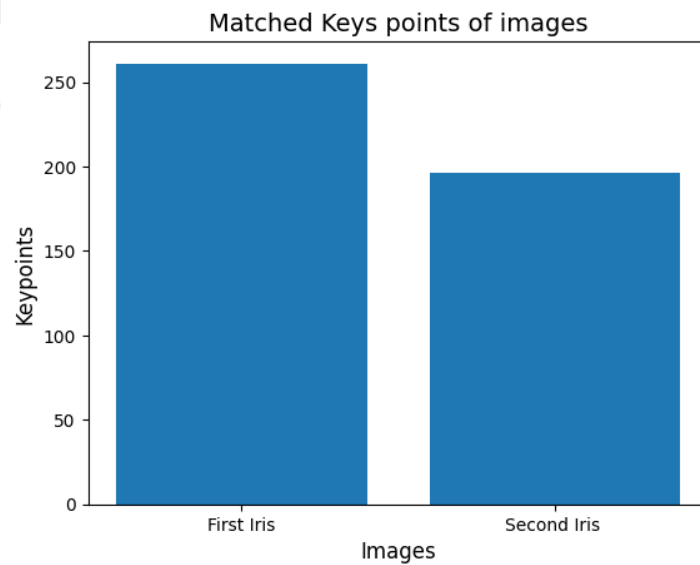


Figure 3.26 Matched Keys points of different irises

3.2.7 Iris Classification

Recent research illuminated that deep convolutional neural network models have become a very robust tool in many applications. For instance, one of the foremost well-known applications of deep learning is image classification [151] [152]. The model applies a convolution neural network, in order to identify unique features in

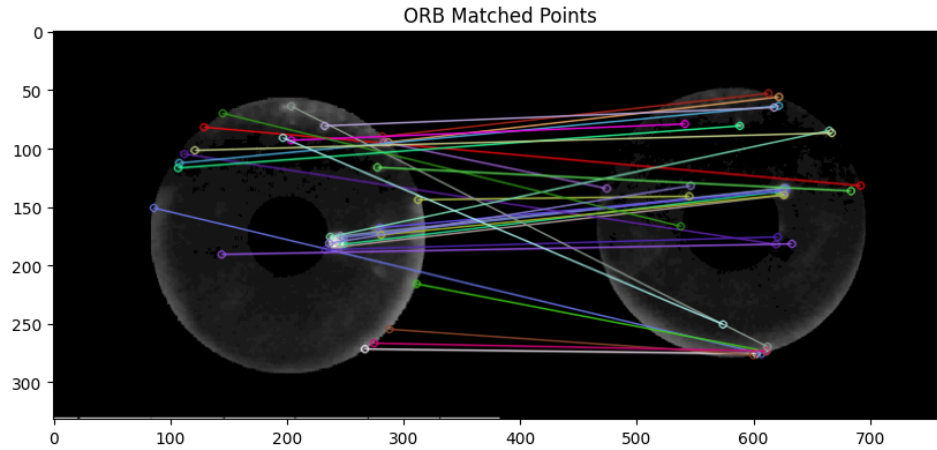


Figure 3.27 ORB Matched Points on different irises

images [153]. Furthermore, in iris identification, the model uses convolutional neural networks. The model uses convolutional neural networks(CNN) for iris identification whereas CNN desire to differentiate between different classes of iris images. The steps that will be done in iris classification, are together the data and making them one, after then preprocessing the data, furthermore, create the model, next time to fit the created model, more importantly, evaluate the model, regulate and formulate different parameters, and improving the model, repeat until satisfied. The most incomparable, initial, and uttermost significant steps in machine learning and deep learning are creating training and a test set. Then the forward step is to turn the data into batches. A batch can be a small subset of the dataset a model looks at through training, for example, rather than observing 20,000 images or even more at one time and trying to discover the patterns, it could be better that a model only looks at 32 images at a time [139]. For creating the model in this research ImageNet is performed. A naive experiential for computer vision models is using the model architecture which is performing best on ImageNet [154], and the architecture is:

Input $\rightarrow$ *Conv + ReLU layers(non - linearities)* $\rightarrow$ *Pooling layer* $\rightarrow$ *Fullyconnected(denselayer)asOutput*

The details about the components of the Conv2D layer used in this research, the 2D intends that the image from inputs is two dimensional, however, those images contain 3 color channels, and the convolutions can be performed on every channel individually, filters are the numbers of feature extractor which are stirring over the images, the kernel

size indicates the size of the filters used, for instance, a kernel size of (3,3) represents that each filter will have the size 3×3 . The stride is the number of pixels a filter will move across as it covers the image, padding adds zeros to the outside of the image so the resulting output of the convolutional layer is the same as the input [139]. After a model is created and compiled, it needs to be fitted, and here two new parameters can be noticed. Steps per epoch which can be the number of batches a model will go through per epoch, in this case, the model goes through all batches so it is equal to the length of train_data and validation_data.

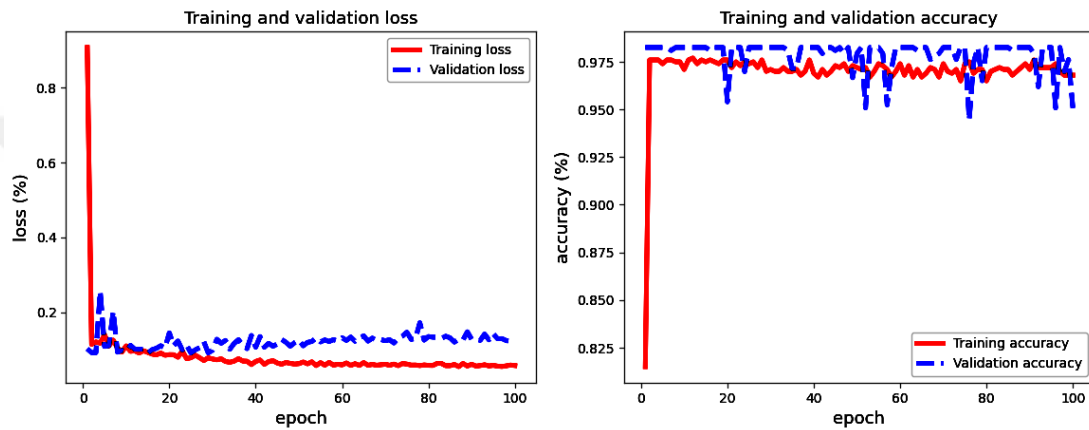


Figure 3.28 The accuracy and loss curves of model's training performance

Figure 3.28 illustrates the proposed CNN model's training which was implemented, and the accuracy and loss curves of the CNN model, the model was performed with 3 convolution-pooling blocks. As it is clear from the model, there can be seen a huge number of trainable parameters almost (906,841), which indicates that it could be sufficient to overfit the data. The accuracy and loss curves clearly represent that the model overfits the data. There's a steady decrease in training loss, and there's a miniature increase in validation loss. One of the best techniques for reducing overfitting is to increase the size of the training dataset [133], when the size of the training data is insignificant and small, then the network tends to have more control over the training data. Other techniques include adding dropout, adding weight decay, and using spatial dropout.

After adding dropout to the model, the validation accuracy obtained in the same number of epochs has improved already as illustrated in Figure 3.29. However, there is still a tiny overfitting occurring.

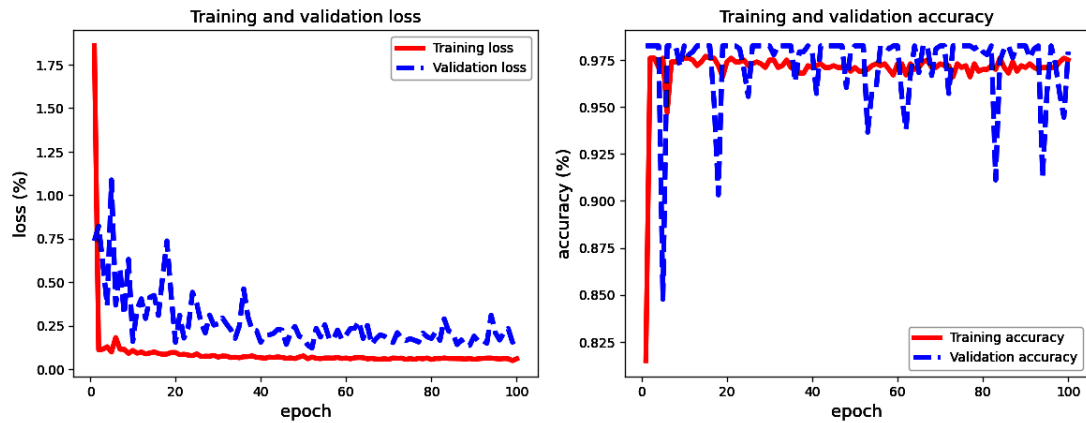


Figure 3.29 The accuracy and loss curves of model's training performance

In this chapter, first of all, CCN iris identification and CNN important layers like Convolution layers, Pooling layers, and Fully Connected Layers have been discussed. Moreover, Iris Identification Phases include iris image acquisition, iris image preprocessing, iris image segmentation, iris image normalization, iris feature extraction, iris pattern matching, and iris image classification. In iris image acquisition some iris databases for instance Ubiiris Version 2 (v2), Multimedia University (MMU), and Gaze Interaction for Everybody (GI4E) which have been used in this research, have been discussed. Furthermore, iris image preprocessing has been performed with a Gaussian filter, Hough transforms to detect iris inner and outer boundaries and edge detection like canny edge detection, ellipse fitting, and sobel edge detection. Then iris image segmentation approaches are done which is performed with U-Net Iris Image Segmentation. Additionally, the iris images have been normalized with Daugman Normalization which proposed a homogeneous rubber sheet model. Moreover, a feature extraction technique based on the Histogram of Oriented Gradient (HOG), VGG – 16 and VGG – 19 Models, and ResNet Models has been applied. Then Iris Pattern Matching based on the ORB algorithm and finally iris image classification has applied.

CHAPTER 4

Experimental Results and Discussion

In this chapter Experimental and analysis of this research have been discussed, and the result, the accuracy, the loss, and the time efficiency of the proposed Iris identification model based on a Deep Convolutional Neural Network were investigated. I have performed a few experimentation and attempts to evaluate the variation and performance of the proposed iris identification and the results of all models have been used in this research. The strength agility of the proposed iris identification is also assessed on various heterogeneous iris image datasets and databases. Previously it was discussed that the data acquisition has not been done by me, in any identification technology the primary phase is image acquisition. This step is very complex and complicated since the color and size of the iris of every person are unique. Due to the different lighting efficacy, the acquisition process might sometimes generate disparate results. I conducted immense implementations using 4 iris datasets, particularly they are Ubiris Version 2 (v2), Multimedia University (MMU), and Gaze Interaction for Everybody (GI4E).

4.1 Iris Databases

Currently, there are eight or even more permitted utilizable iris image datasets that are free to the public or academic use for iris identification biometric purposes, the detailed description of the datasets is given in the following sections.

4.1.1 Ubiris Version 2 (v2)

Ubiris v2 is the extended version of ubiris v1. this dataset is an open and free utilizable UBIRIS.v2 version of the database that contains over 11000 images from 241 subjects, and continuously increasing, and more pragmatic noise factors, the dataset contains two sessions. Furthermore, for the equivalent reasons in ubiris v1, just session 1 was used in ubiris version 2. The essential Purpose of the UBIRIS database is to diminish the necessity of operator cooperation. The images were accumulated via a Nikon Canon

EOS5D camera and the focal length = 400 and image resolution = 200×150 pixels [130], images were indeed captured at a distance and on the move).

4.1.2 Multimedia University (MMU)

The Multimedia University (MMU) [128], has introduced two iris databases. The primary database is MMU1 which contains 450 images, taken by a semi-automated camera devoted to iris capturing (LG IrisAccess 2200) at a 7-25 cm distance range, and the next database is MMU2, comprising 995 images, composed by Panasonic BM-ET100US camera at a distance range of 47- 53 cm from the human subject, the iris images in MMU database [128] have been collected from 100 volunteers of different ages and different nationalities, with each volunteer contributing 5 images from each eye.

4.1.3 Gaze Interaction for Everybody (GI4E)

Gaze Interaction for Everybody (GI4E) [132]: The database is public and images have been interpreted via three different experts, for instance, eye corners and iris centers. Images are classified into categories, The classifiers are measured in corner and center categories. The dataset contains 1236 images from 103 different persons aged from 18 to 83 years old, males and females. As a result, it can be understood that from each user 12 images are captured. The set of images was accumulated using a low-cost webcam, with automatic lighting correction, in which the image size is 800x600 pixels [132]:. The goals of this database are eye-tracking technology and to identify the inner iris part, Gi4E database is a more pragmatic database for the sake eye tracking attempts. It encompasses images with head and eye movements.

4.2 Implementation and Experimental Results

The iris identification experiments were implemented on images from the UBIRIS v2, and Multimedia University (MMU) Gaze Interaction for Everybody (GI4E). After preprocessing, segmentation, and further iris image feature extraction, additionally, iris images were trained with the convolutional neural network models, for effective

and impressive iris classification. The preprocessing includes two sections, iris image filtering which involves noise reduction, color normalization, edge detection, and histogram equalization, and iris edge detection, which finds the boundaries of substances through the images, it detects interruptions in brightness, iris edge detection also contains three phases such as iris inner boundary localization, outer boundary localization, and Localization of boundary between eyelids and iris [29]

4.2.1 Evaluation of the Iris Preprocessing Scheme

In this research image preprocessing has been done through the Gaussian filter, Mean filter, and Canny edge detection, Hough Transform, Ellipticity edge detection. image filtering is the first step in iris image preprocessing and is performed previously then the edge detection, which is to omit the intrusion and interference in the original image for the edge detection. The important goal of an image preprocessing filter involves Noise Reduction, Color Normalization, Edge Detection, and Histogram Equalization, and iris image preprocessing filters are primarily used to support the high frequencies in the image or the low frequencies in the image, Figure 4.1 illustrates the Histogram Equalization before and after image filtering. As Image (a) illustrates the Histogram before and image (b) shows the Histogram after the image exponential operator which $r=0.45$, see equation (3.3). After the exponential operator, I implemented the histogram equalization operator on eye images from the database (UBIRIS v2). The intensity of the pixel is specified $i \in [0..255]$, the image histogram h , the total number of pixels of the image N , compute the cumulative distribution function in equation (3.7). In Figure 4.1 image (c) and image (d) illustrate the implementation of the function and plot the histogram equation of the given image before and after the transformation. In this research first the mean filter and then the Gaussian filter has been applied.

Furthermore, edge detection has been performed, in this thesis, Hough transform, Canny edge detection, Sobel edge detection, and RANSAC Ellipse Fitting. Hough Transform is essential to be used in case of detecting lines, circles, or any kind of parametric curves, in this thesis Hough transform is used to detect the iris circle as demonstrated in Figure 4.2, image (a) shows the input image from UBIRIS v2 database, image (b) is the color image which Hough transform detected the iris and image (c) is

the grayscale iris outer and inner boundary detection by Hough transform.

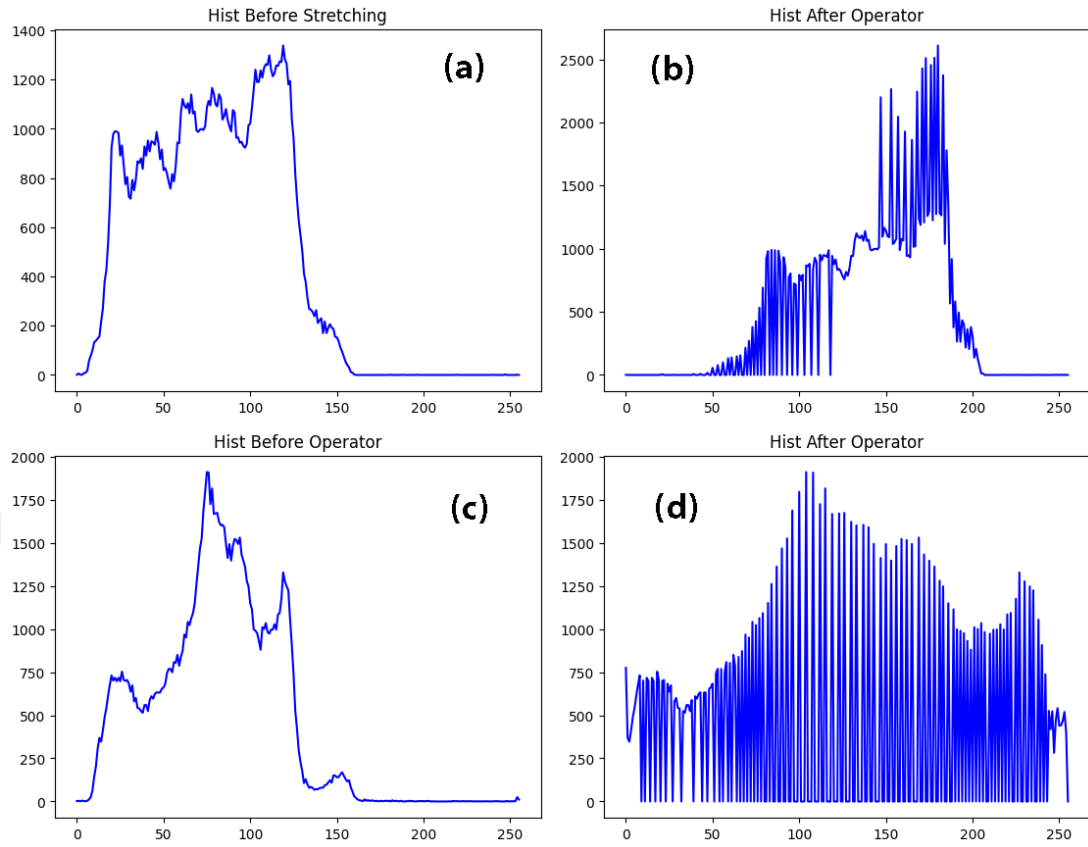


Figure 4.1 Histogram Equalization before and after image filtering

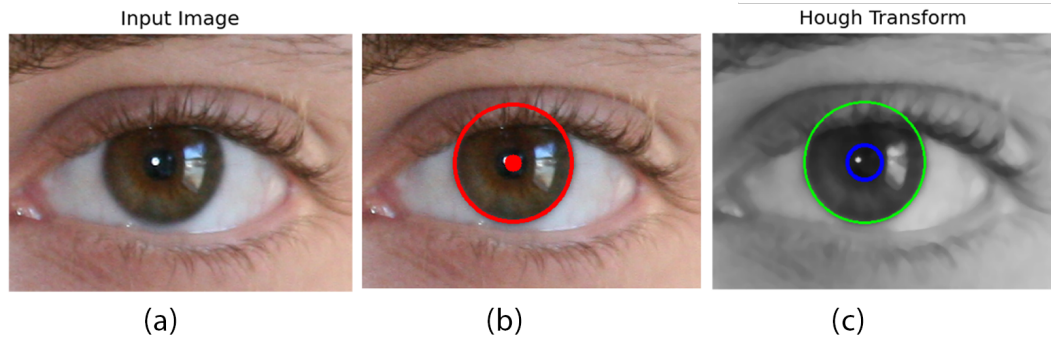


Figure 4.2 Hough Transform iris outer and inner boundary detection

4.2.2 Accuracy and Evaluation of the Iris Segmentation Scheme

Many experiments were performed on the mentioned iris datasets to validate the proposed iris image segmentation scheme's performance. Some different segmentation methods were implemented to segment the iris images. The iris images were segmented through Hough Transform, U-Net, and Daugman's models. To add an important point is that, the proposed classification algorithms applied for the iris

identification process were also applied after segmentation and other stages as well, in order to patronize the defend an identical measure in the evaluation schedule. In this research study for the aim of iris segmentation, the U-Net model illustrated logical performance in areas of segmentation. The architecture of the U-Net model in this learning with an iris image size of 240×320 pixels as an input has been used. Furthermore, for Conv 1, two convolution operations by a 3×3 convolutional kernels were executed, with the size of 256×256 pixels the number of feature maps is 64, and alongside the max-pooling with a kernel size of 2×2 was performed. In U-Net segmentation model training after 150 epochs attaining 94.98% accuracy on the training set and 93.34% on the test set, and this amount is sufficient because this is the first time, I have used the U-Net model to segment the iris images. Four thresholds with the numbers 0.1, 0.2, 0.3, and 0.4 have been used, and the outcome is demonstrated in Figure 4.3. And as well Figure 4.4 illustrates the accuracy and loss curve of the training and test set on the U-Net model.

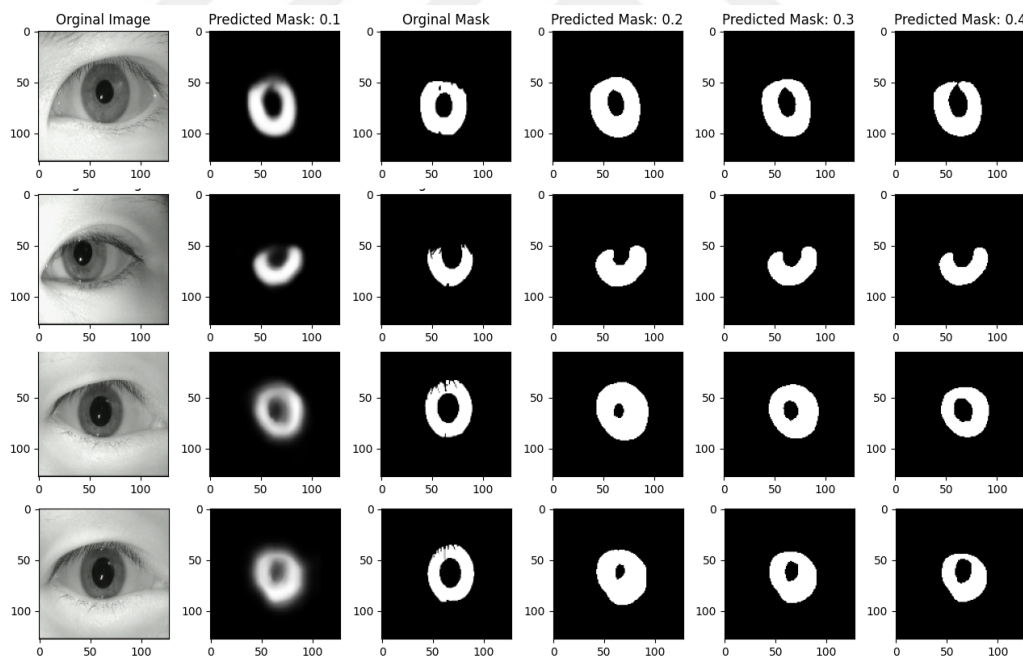


Figure 4.3 Iris Segmentation result on U-Net model

As it is clear from Figure 4.3, it mainly extracts foreground based on gray value information, so it is especially useful for the segmentation of images with strong contrast between foreground objects and background, and the comparison of the segmentation results of the proposed algorithm and the ground truth mask. Figure 4.5 demonstrates

the Iris Segmentation accuracy and loss graph on the CNN Model.

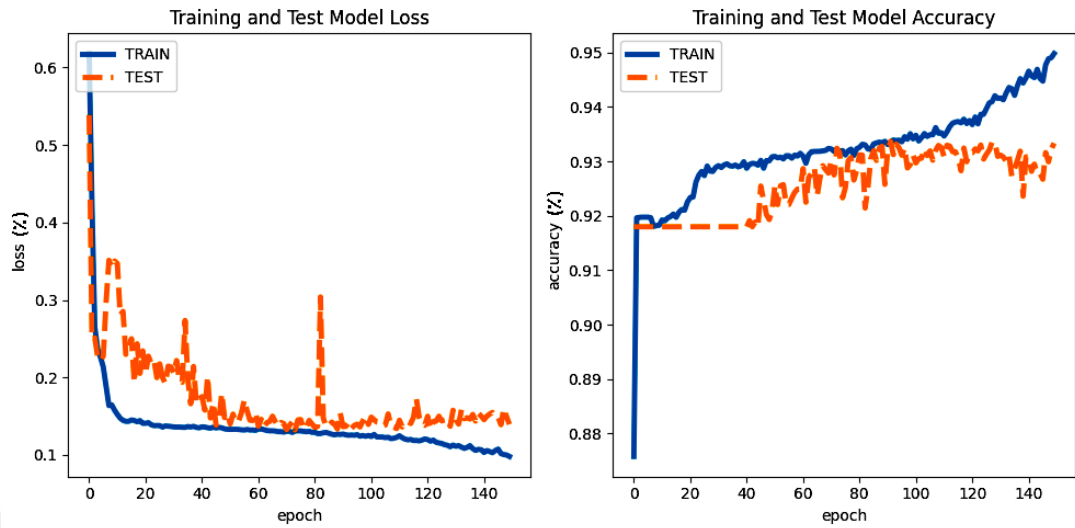


Figure 4.4 Iris Segmentation accuracy and loss curve with U-Net model

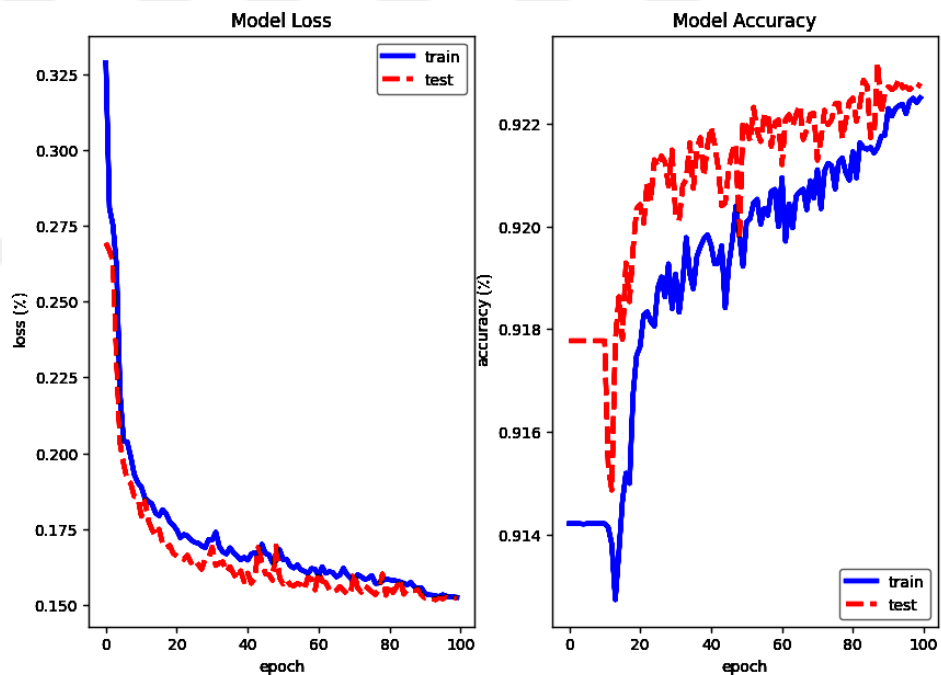


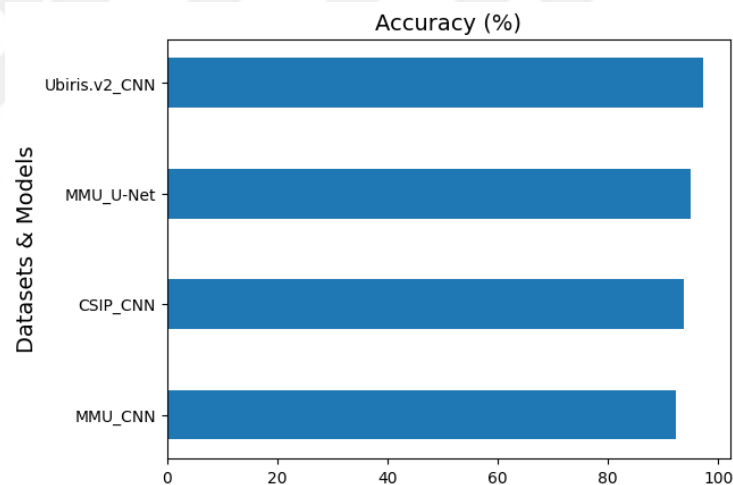
Figure 4.5 Iris Segmentation accuracy and loss graph with CNN model

Table 4.1 demonstrates the segmentation result between MMU, CSIP, and Ubiris.v2 datasets and the comparison between the performance of models illustrated, as we can see from the mentioned table, accuracy, and error rate. standard deviation and mean have been calculated,

Table 4.1 Iris Segmentation Model Training Comparisons and the Results (%)

Dataset	Model	ACC	EER	STD	Mean
MMU	CNN	92.25	0.1525	0.6464	0.2132
MMU	U-Net	94.98	0.0980	0.6556	0.2135
CSIP	CNN	93.80	0.1294	0.6419	0.2128
Ubiris.v2	CNN	97.30	0.1111	0.6317	0.2751

Furthermore, after training the model for iris segmentation with the CNN model I have used the U-Net pre-trained model of CNN with kernel_size of 3×3 , and the same as sample CNN using activation function (Relu and Sigmoid) and binary cross-entropy as training objective and adaptive moment estimation (Adam), and four thresholds (0.1, 0.2 0.3 and 0.4) have been used, the number of epochs is 150 and batch_size is 32, and the accuracy result is illustrated in Figure 4.6 and also demonstrates the iris segmentation accuracy comparison graph between CNN model and U-Net model with Ubiris.v2, CSIP, and MMU databases.

**Figure 4.6** Iris Segmentation accuracy Comparison graph

Additionally, in Figure 4.7 the errors through the CNN and U-Net models have been illustrated which have been mentioned in Table 4.1, and after that, the numerical percentage of the errors was calculated best of the proposed algorithms I obtained the standard deviation of the iris image segmentation, which is shown in Figure 4.8.

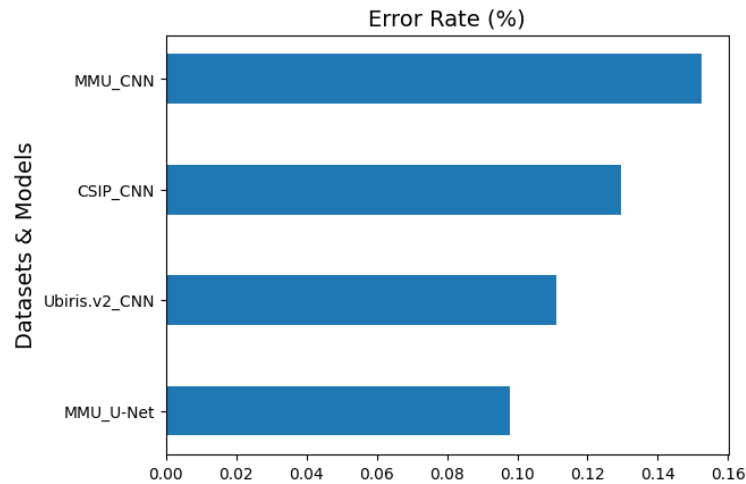


Figure 4.7 Iris Segmentation Error Rate Comparison graph

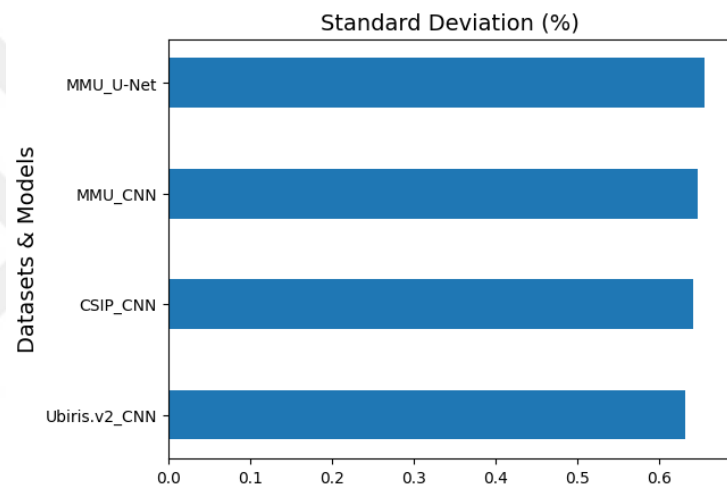


Figure 4.8 Iris Segmentation Standard Deviation Comparison graph

4.2.3 Feature Extraction Evaluation Scheme

To accredit the iris feature extraction methodology, the iris images from the iris datasets are first preprocessed and then segmented using the CNN U-Net iris segmentation algorithm and then normalized in order to attain advanced accuracy. As was discussed in Chapter 3, I performed a Histogram of oriented gradients (HOG), VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet, and InceptionV3 to extract the iris features from the images. An experiment was performed with just 10 percent of images from the training data set in feature extraction, it looks like after the first experiment there is overfitting in the model. The accuracy and loss graph obviously represent that the model overfits the data. There's a consecutive decrease in training loss, and there's a small increase in validation loss. One of the greatest methods for reducing overfitting

is to increase the size of the training dataset [155], when the size of the training data is insignificant and small, then the network tends to have more control over the training data. Other techniques are to adding dropout, adding weight decay, and using spatial dropout. The ResNet50 V2 feature extraction, model was able to attain around 97.62% accuracy on the training data and almost 98.35% accuracy on the test data. That tends to express the strength of deep learning, CNN, and transfer learning. Figure 4.9 elucidates the result of ResNet50 V2.

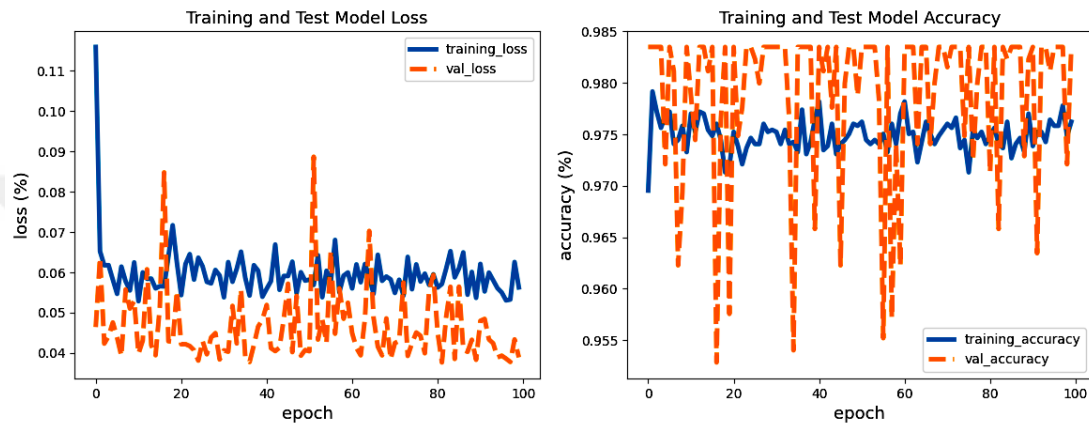


Figure 4.9 Model training accuracy and loss result on ResNet50 V2

The feature extraction layer has 23,568,898 parameters which are pre-learned patterns, we can say that the model has already learned on the ImageNet dataset [155]. Since in this Research, the set trainable parameter is set to False, these layouts stay frosty during the training. From the result, it can be derived that during training the model updates the 4,098 trainable params in the output layer to assort the database. It has been trained a ResNet50 V2 model, now it is the moment to perform the identical with the EfficientNetB0 model and add the dropout in the model. The EfficientNetB0 model does even better than the ResNet50 V2 model [156]. Obtaining 97.68% accuracy on the training data and 98.35% accuracy on the test data, same as ResNet50 V2 with just 10% of the data from the iris dataset.

After adding dropout to the model, the validation accuracy was obtained in the same number of epochs and almost the same accuracy, and there is a great decrease in validation loss, which I was looking for to solve, Figure 4.10 illuminates the result in EfficientNetB0.

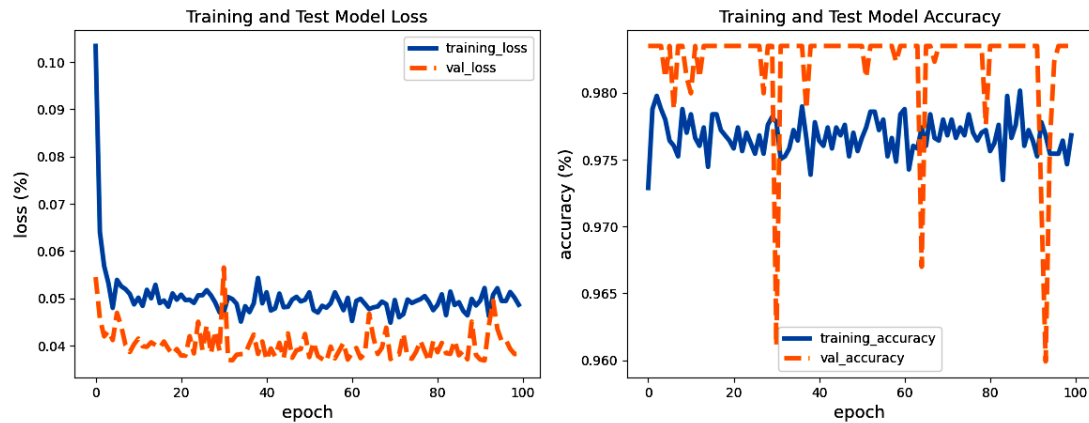


Figure 4.10 Model training accuracy and loss result on EfficientNetB0

Table 4.2 represents the iris image feature extraction comparisons and the results between MMU and Ubiiris.v2 datasets, and also the comparison between the performance of CNN, ResNet50 V2, and EfficientNetB0 models was illustrated, as can be seen in the mentioned table, accuracy, error rate. standard deviation and mean have been calculated,

Table 4.2 Feature Extraction Model Training Comparisons and the Results(%)

MTS	Dataset	Model	ACC	EER	STD	Mean
Training Set	UBIRIS v2	CNN	96.30	0.019	0.3903	0.3700
Test Set	UBIRIS v2	CNN	98.25	0.148	0.3667	0.3451
Training Set	MMU	ResNet50 V2	97.62	0.056	0.3754	0.3979
Test Set	MMU	ResNet50 V2	98.35	0.038	0.3722	0.4029
Training Set	MMU	EfficientNetB0	97.68	0.048	0.3718	0.3929
Test Set	MMU	EfficientNetB0	98.35	0.038	0.3686	0.3977

As I mentioned perversely, in this research, I have used several models to extract iris features, additionally, I perused the susceptibility and capabilities of a convolutional neural network comprehended using the VGG16 and VGG19 methods, mostly known as a convolutional community model. And it is obvious from Table 4.2, in iris feature extraction all the models have obtained a great result.

Figure 4.11 illustrates the iris image feature extraction accuracy and the comparison graph between the CNN model. ResNet50 V2 model and EfficientNetB0 model with Ubiris.v2, CSIP, and MMU datasets. and furthermore, in Figure 4.12 the errors throughout the CNN, ResNet50 V2, and EfficientNetB0 models have been illustrated which have been calculated in iris image feature extraction and the comparison results are stated in Table 4.2.

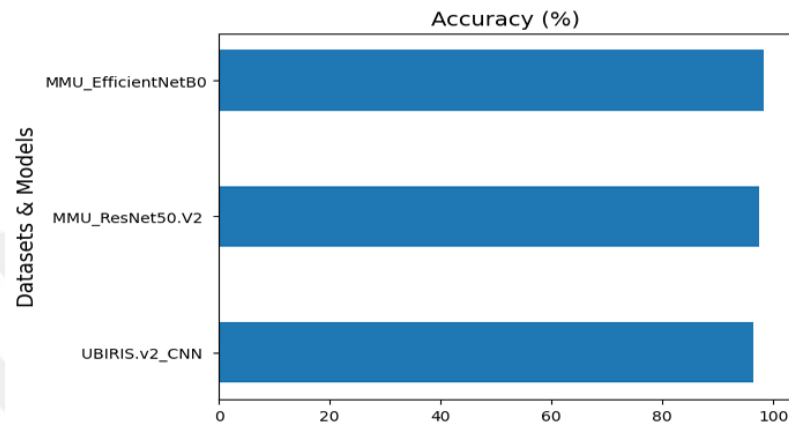


Figure 4.11 Iris Image Feature Extraction Accuracy Comparison graph

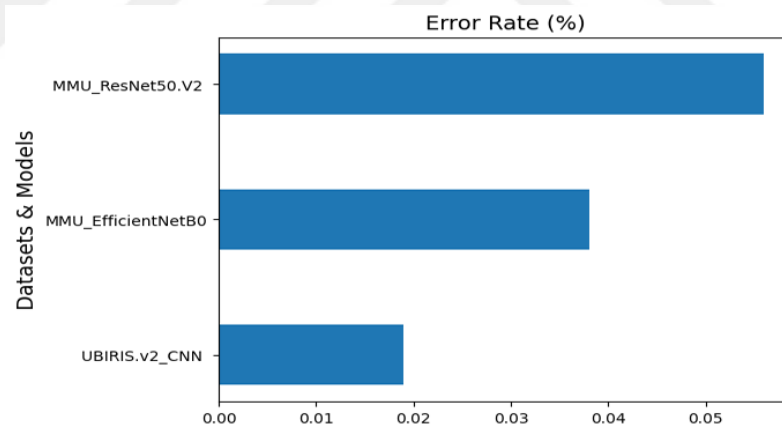


Figure 4.12 Iris Image Feature Extraction Error Rate Comparison graph

Finally, in iris image extraction, the errors were calculated through CNN, ResNet50 V2, and EfficientNetB0 models, which obtained the standard deviation of the iris image feature extraction, which is presented in Figure 4.13.

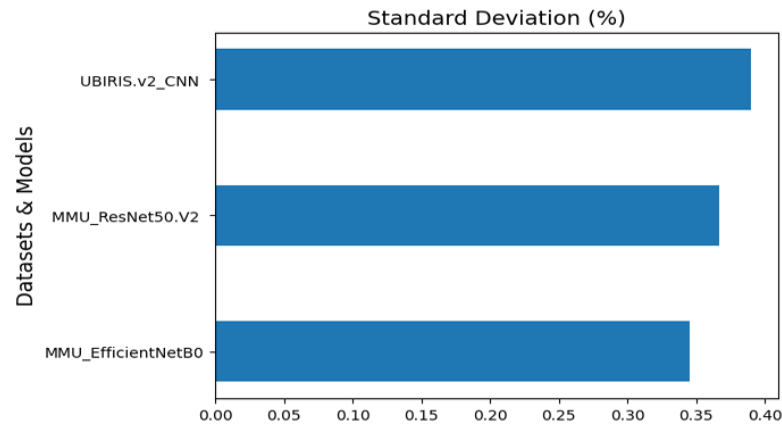


Figure 4.13 Iris Image Feature Extraction ST Comparison graph

4.2.4 Evaluation of the Iris Classification scheme

In iris identification, I performed the convolutional neural network (CNN) model, whereas CNN desired to differentiate between different classes of iris images. First, I created a training and a test set which is very significant for training a model, after then time to turn the data into batches. First, the model was performed with 3 convolution-pooling blocks. Figure 4.14 illustrates the model's training performance and the accuracy and loss curves of the CNN model.

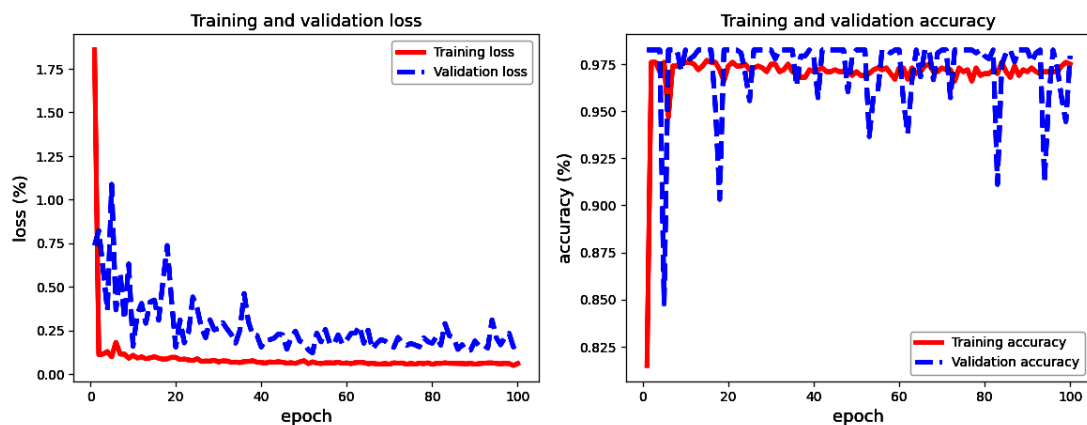


Figure 4.14 The accuracy and loss curves of model's training performance

Then ImageNet was performed to create the model [157]. After a model is created and compiled, it has been fitted, which obtained 98.11% accuracy on the training data set and 98.35% on the test data set, and the result is indicated in Figure 4.15 on ImageNet. Furthermore, I have created a CNN model on TinyVGG with a 3-layer Convolutional Neural Network, achieving 98.11% accuracy on the training data set and 98.35% accuracy on the test data set on TinyVGG. These two models performed the

same in this research. Figure 4.16 demonstrates the model's training performance on TinyVGG and the accuracy and loss curves.

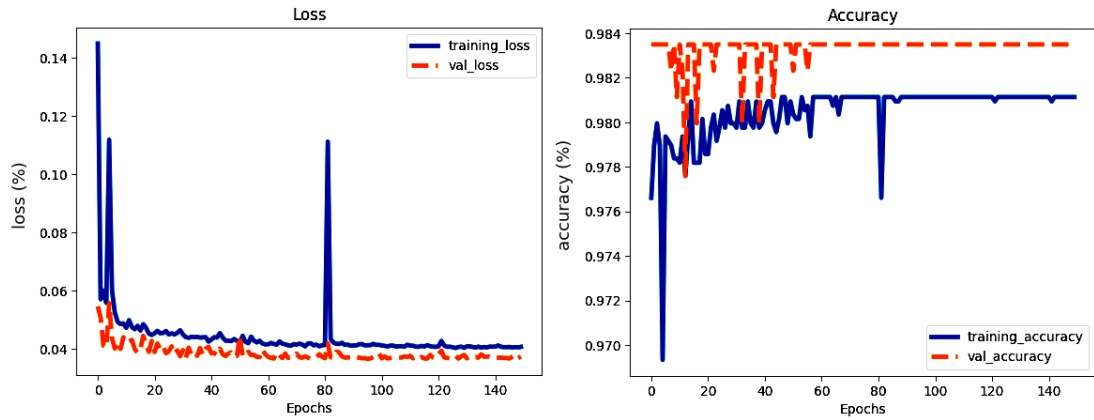


Figure 4.15 ImageNet accuracy and loss performance

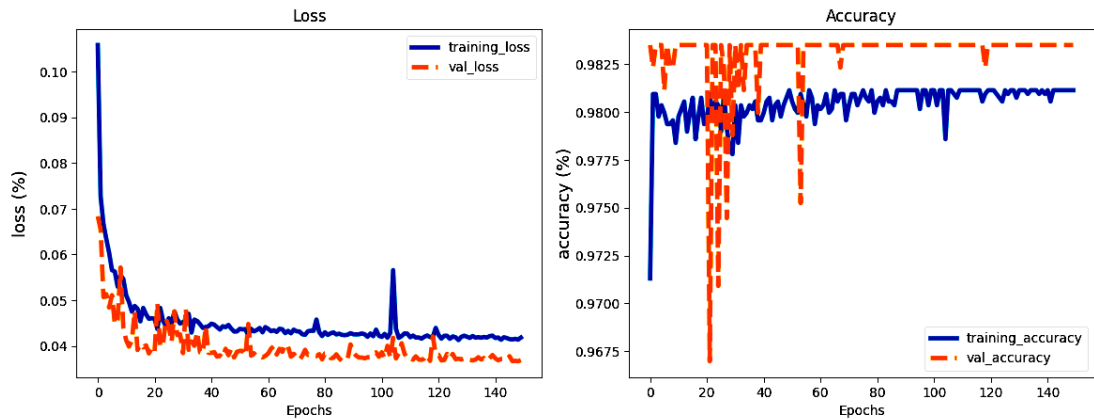


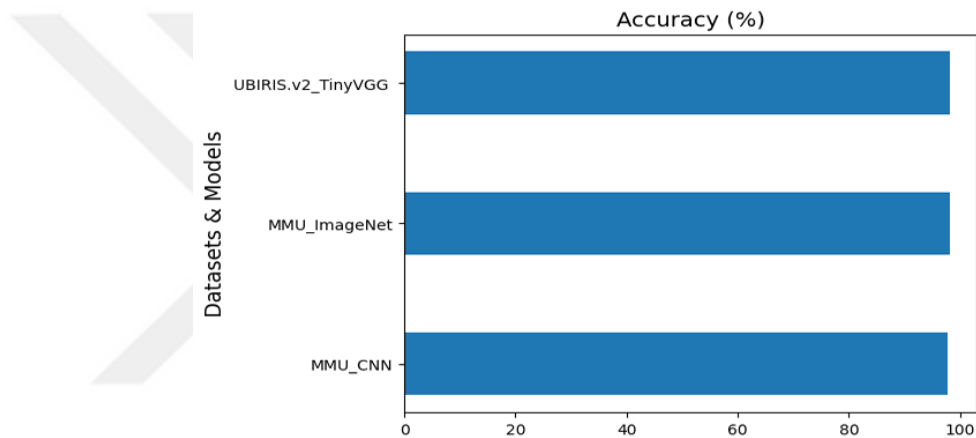
Figure 4.16 TinyVGG accuracy and loss performance

Table 4.3 indicates the iris image classification comparisons and the results between MMU and Ubiris.v2 datasets, and also the comparison between the implementation and efficiency of CNN, ImageNet, and TinyVGG models which were illustrated. Actually, the ImageNet and TinyVGG typically follow the CNN structure and achieved the same accuracy results. As it is clear from the mentioned table, accuracy, and error rate. standard deviation and mean have been calculated.

Table 4.3 Image Classification Comparisons and the Results (%)

MTS	Dataset	Model	ACC	EER	STD	Mean
Training Set	MMU	CNN	97.50	0.059	0.3704	0.3733
Test Set	MMU	CNN	97.94	0.172	0.3739	0.3718
Training Set	MMU	ImageNet	98.11	0.040	0.3708	0.3712
Test Set	MMU	ImageNet	98.35	0.148	0.3710	0.3699
Training Set	UBIRIS v2	TinyVGG	98.11	0.041	0.3708	0.3712
Test Set	UBIRIS v2	TinyVGG	98.35	0.036	0.3691	0.3656

Figure 4.17 illustrates the iris image classification accuracy and the comparison graph between the CNN model, ImageNet, and TinyVGG pre-trained models, as the accuracy percentages are explained in Table 4.3.

**Figure 4.17** Iris Image Classification accuracy Comparison graph

Moreover, in Figure 4.18 the errors through the CNN, ImageNet and TinyVGG models have been demonstrated which have been calculated in iris image classification the comparison results are stated in Table 4.3, and after that, the errors were calculated through CNN, ImageNet, and TinyVGG models, which obtained the standard deviation of the iris image classification, which is presented in Figure 4.19.

In the Experimental Results and Discussion chapter, I trained and evaluated the proposed iris identification system with various iris image datasets, and the details about the selected database have been discussed. Several experiments were implemented on the mentioned iris datasets to validate the selected iris image segmentation scheme's performance. CNN and U-Net segmentation approaches were implemented on iris images and are segmented using: Hough Transform, U-Net, and Daugman's model. The U-Net segmentation model training after 150 epochs attained

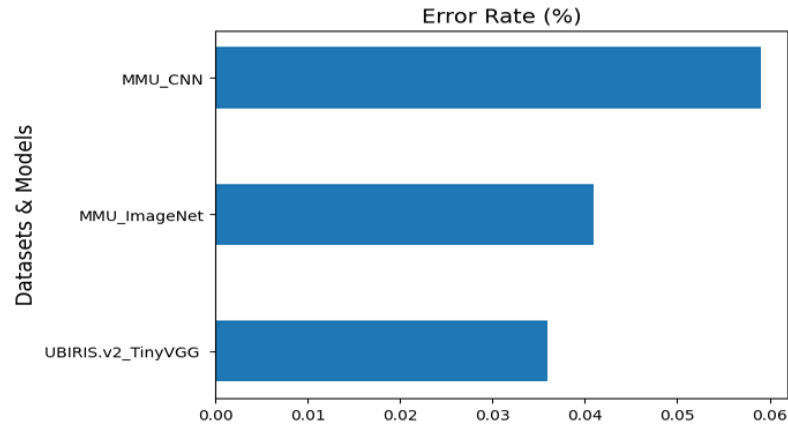


Figure 4.18 Iris Image Classification Error Rate Comparison graph

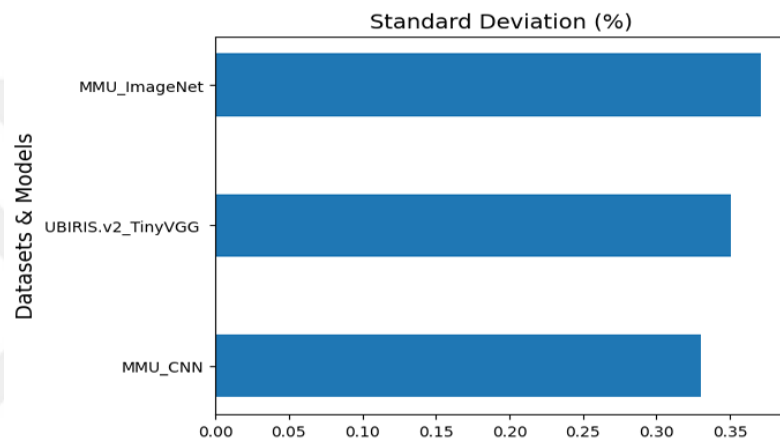


Figure 4.19 Iris Image Classification Standard Deviation Comparison graph

94.98% accuracy on the training data set and 93.34% on the test data set. Four thresholds with the numbers 0.1, 0.2, 0.3, and 0.4 have been used. To accredit the iris feature extraction methodology, the iris images from the iris datasets are first preprocessed and then segmented using the CNN U-Net iris segmentation algorithm and then normalized in order to attain advanced accuracy. I performed a Histogram of oriented gradients (HOG), VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet, and InceptionV3 to extract the iris features from the images. An experiment was performed with just 10 percent of images from the training data set in feature extraction, it looks like, after the first 100 epochs, the ResNet50 V2 feature extraction, model was able to attain around 97.62% accuracy on the training data set and almost 98.35% accuracy on the test data set. Then EfficientNetB0 model was also implemented and achieved the same accuracy. Finally, an Evaluation of the Iris Classification with a convolutional neural network (CNN) was performed, whereas

CNN desired to differentiate between different classes of iris images.



CHAPTER 5

Conclusion

In this concluding chapter, I summarize the key contributions of this research. With the imminent need for security and along with the increasing request for information security, security regulations, and trustworthy method of identification and adjustment in biometric systems all over the world, biometric identification technology transpires extensively used in our regular life. In this research, a new biometric identification system is proposed, which is based on a deep learning algorithm for identifying humans using biometric modalities of the iris. Furthermore, machine learning algorithms necessitate and involve certain preprocessing stages to be derived before the feature extraction. In the interpretation of the countless implementation of CNN, and deep learning algorithms in several identification obligations, the structure and establishment of the proposed iris identification are based on deep convolutional neural networks (CNNs) which have the ability to extract features from images and classify those images. This research proposed a new iris identification that achieves very high accuracy on different public datasets based on convolutional neural networks. Namely, the datasets are UBIRIS Version 2.0, Multimedia University (MMU), and Gaze Interaction for Everybody (GI4E) datasets. Accurate image preprocessing and iris image segmentation play a significant task in iris identification biometrics. In this research first, the iris images were preprocessed for the sake of noise reduction, Color Normalization, Edge Detection, Histogram Equalization, and detected the boundary of the pupil. The objective of This approach is clearly to reduce and remove the noise from iris images, augment the data, and abridge the information and data in order to obtain an arbitrary iris image with a stable region and high quality. In areas such as image preprocessing, machine vision, and computer vision edge detection are commonly utilized to segment the images or data and extract features from data. Most of the figures and shapes information of an image is surrounded by edges

This research proposed a new iris segmentation method model for iris segmentation, namely they are CNN and U-Net that raised the efficacy of final accuracy in training

and testing data. The architecture of the U-Net model in this research with an iris image size of 240×320 pixels as an input has been used. The feature maps with (width x height) dimension along on the left represented by Individual box. Designed for Conv 1, two convolution operations by a 3×3 convolutional kernels were performed, with the size of 256×256 pixels the number of feature maps is 64, and alongside the max-pooling was performed with a kernel size of 2×2 . Iris images might be taken in several sizes and with different distances, as a result of brightness differences, the circular and radial size of the pupil may vary appropriately. Thus, the iris area has to be normalized to reimburse for this alteration. The proficiency and implementation of iris identification systems particularly depend on segmentation and normalization. The normalization process normalizes iris images to a constant dimension and sizeable fragment to avoid its incompatibility that might arise for various reasons such as iris imaging remoteness discrepancy, a revolution of the device, illumination variation, iris stretching, eye rotation, head deformation and etc. Therefore, iris image normalization is applied to decrease all of these obstacles.

I performed a Histogram of oriented gradients (HOG), VGG-16, VGG-19, ResNet-50 vs ResNet101, MobileNet, AlexNet, and InceptionV3 to extract the iris features from the images. An experiment was performed with just 10 percent of images from the training data set in feature extraction, it looks like, after the first 100 epochs, the ResNet50 V2 feature extraction, model was able to attain around 97.62% accuracy on the training data set and almost 98.35% accuracy on the test data set. Then EfficientNetB0 model was also implemented and achieved the same accuracy. Furthermore, in iris identification, the model uses a convolutional neural network (CNN). The model uses a convolutional neural network (CNN) for iris identification whereas CNN desire to differentiate between different classes of iris images. In iris identification, I have created a training and a test set which is very significant for training a model, and then time to turn the data into batches. First, the model was performed with 3 convolution-pooling blocks. After a model is created and compiled, it has been fitted, which obtained 98.11% accuracy on the training data set and 98.35% on the test data set. And furthermore, I have created a CNN model on TinyVGG with a 3-layer Convolutional Neural Network, achieving 98.11% accuracy

on the training data set and 98.35% accuracy on the test data set on TinyVGG. These two models performed the same in this research.

Future Work: In this research, the proposed iris identification approaches perform reasonably well in degraded iris images. However, there are numerous matters that should be addressed and resolved in order to improve the performance even more. Future research could be conducted along the following directions to build a robust iris (biometrics) identification system.

1. In this research, the CNN models, Hough Transform, Daugman uses the combination of the strengths of the gradient local and region-based methods for finding the iris boundaries. However, a simple region-based method can still be used by fusing multiple regions in order to determine the optimum boundaries of the iris. Iris identification biometric technology remains one of the most promising areas of biometric access control in the global market. Other than the selected criteria, there are still lots of other outstanding published networks such as Inception, GoogleNet, and NaSNet-Large present. This research can be extended to adapt and test other available models on the different public iris datasets. To minimize the trade-off matter, the model reduction technique can be considered to eliminate redundant layers and neurons, thus, decreasing the size of the network.
2. Moreover, reversing the order of the identification process can significantly improve the method. Processing the eyelashes first and eliminating the eyelid/eyelash before applying the segmentation, can greatly enhance the segmentation accuracy and act as a way of accessing the quality of the original iris image before initializing the processing of the image.
3. Meanwhile the quality of the images affects the overall matching accuracy and in order to eliminate time wasted in highly degraded iris images, an iris image quality assessment scheme needs to be employed to eliminate closed eyes and another iris image that cannot pass for recognition. Most current iris image quality assessment methods deal exclusively with the iris images captured in a Near Infra-Red (NIR) setup, an enhanced method can be deployed to assess the quality of iris images

captured by using the Visible Wavelength (VW) light imagery. In general, the VW imaging setup is able to acquire iris data at significantly larger distances and on moving subjects.

4. In this research, I have dealt with iris images that have been captured in both NIR and VW imagery setups. Though, a more sophisticated iris segmentation scheme can be developed for the VW iris images only. In this way, the whitish sclera can be utilized for segmentation instead of the iris information.
5. The iris liveness detection is a major issue in the area of iris identification otherwise, a high-resolution photograph can be presented to an iris identification system which might result in a false match. Fake iris detection is another important factor that should be given careful attention. The contact lenses vastly used nowadays are capable of changing the color of an individual's iris. This may lead to problems with the iris recognition system since a fake iris pattern is printed on the surface of the lens. The system may lead to false rejection of an enrolled user. Contrary, the system may falsely accept a subject, if the iris patterns of that subject have been enrolled in the database. Therefore, further research is required to overcome these challenges.

REFERENCES

- [1] Panasiuk P., Szymkowski M., Dąbrowski M., and Saeed K., “A multimodal biometric user identification system based on keystroke dynamics and mouse movements,” in *Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings 15*. Springer, 2016, pp. 672–681.
- [2] Bouzouina Y. and Hamami L., “Multimodal biometric: Iris and face recognition based on feature selection of iris with ga and scores level fusion with svm,” in *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*. IEEE, 2017, pp. 1–7.
- [3] Alay N. and Al-Baity H. H., “Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits,” *Sensors*, vol. 20, no. 19, p. 5523, 2020.
- [4] Zhang Q., Yang L. T., Chen Z., and Li P., “A survey on deep learning for big data,” *Information Fusion*, vol. 42, pp. 146–157, 2018.
- [5] Bertillon A., “la couleur de l’iris, revue scientifique,” 1885.
- [6] Masek L. *et al.*, “Recognition of human iris patterns for biometric identification,” 2003.
- [7] Ukpai C. O., “Biometric iris image segmentation and feature extraction for iris recognition,” Ph.D. dissertation, Newcastle University, 2016.
- [8] Patil P. M., “Iris recognition in less constrained environment,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 7, pp. 196–200, 2013.
- [9] He Z., Tan T., Sun Z., and Qiu X., “Toward accurate and fast iris segmentation for iris biometrics,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1670–1684, 2008.

- [10] He Z., Tan T., and Sun Z., "Iris localization via pulling and pushing," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4. IEEE, 2006, pp. 366–369.
- [11] Proenca H., "Iris recognition: On the segmentation of degraded images acquired in the visible wavelength," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1502–1516, 2009.
- [12] Bowyer K. W., Hollingsworth K., and Flynn P. J., "Image understanding for iris biometrics: A survey," *Computer vision and image understanding*, vol. 110, no. 2, pp. 281–307, 2008.
- [13] Roy K., Bhattacharya P., and Suen C. Y., "Towards nonideal iris recognition based on level set method, genetic algorithms and adaptive asymmetrical svms," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 458–475, 2011.
- [14] He Z., Tan T., Sun Z., and Qiu X., "Toward accurate and fast iris segmentation for iris biometrics," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1670–1684, 2008.
- [15] Bowyer K. W., Hollingsworth K. P., and Flynn P. J., "A survey of iris biometrics research: 2008–2010," *Handbook of iris recognition*, pp. 23–61, 2016.
- [16] Bowyer K. W., Hollingsworth K., and Flynn P. J., "Image understanding for iris biometrics: A survey," *Computer vision and image understanding*, vol. 110, no. 2, pp. 281–307, 2008.
- [17] Tisse C.-l., Martin L., Torres L., Robert M. *et al.*, "Person identification technique using human iris recognition," in *Proc. Vision Interface*, vol. 294, no. 299, 2002, pp. 294–299.
- [18] Masek L. *et al.*, "Recognition of human iris patterns for biometric identification," 2003.
- [19] Kong W. and Zhang D., "Accurate iris segmentation based on novel reflection and eyelash detection model," in *Proceedings of 2001 International Symposium*

- on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No. 01EX489).* IEEE, 2001, pp. 263–266.
- [20] He Z., Tan T., Sun Z., and Qiu X., “Robust eyelid, eyelash and shadow localization for iris recognition,” in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 265–268.
- [21] Proença H. and Alexandre L. A., “Toward noncooperative iris recognition: A classification approach using multiple signatures,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 4, pp. 607–612, 2007.
- [22] Mansfield T., Kelly G., Chandler D., and Kane J., “Biometric product testing final report,” *Contract*, vol. 92, no. 4009, p. 309, 2001.
- [23] Nguyen K., Fookes C., Ross A., and Sridharan S., “Iris recognition with off-the-shelf cnn features: A deep learning perspective,” *IEEE Access*, vol. 6, pp. 18 848–18 855, 2017.
- [24] Shah S. and Ross A., “Iris segmentation using geodesic active contours,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 824–836, 2009.
- [25] Roy K., Bhattacharya P., and Suen C. Y., “Unideal iris segmentation using region-based active contour model,” in *Image Analysis and Recognition: 7th International Conference, ICIAR 2010, Póvoa de Varzin, Portugal, June 21-23, 2010, Proceedings, Part II 7*. Springer, 2010, pp. 256–265.
- [26] Ma L., Wang Y., Tan T. *et al.*, “Iris recognition based on multichannel gabor filtering,” in *Proc. Fifth Asian Conf. Computer Vision*, vol. 1, 2002, pp. 279–283.
- [27] Daugman J., “Demodulation by complex-valued wavelets for stochastic pattern recognition,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 1, no. 01, pp. 1–17, 2003.
- [28] Roy K. and Bhattacharya P., “Nonideal iris recognition using level set approach and coalitional game theory,” in *Computer Vision Systems: 7th International*

Conference on Computer Vision Systems, ICVS 2009 Liège, Belgium, October 13-15, 2009. Proceedings 7. Springer, 2009, pp. 394–402.

- [29] Daugman J., “How iris recognition works,” in *The essential guide to image processing*. Elsevier, 2009, pp. 715–739.
- [30] Daugman J., “Iris recognition: The colored part of the eye contains delicate patterns that vary randomly from person to person, offering a powerful means of identification,” *American scientist*, vol. 89, no. 4, pp. 326–333, 2001.
- [31] Liu X., Bowyer K. W., and Flynn P. J., “Experimental evaluation of iris recognition,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)-Workshops*. IEEE, 2005, pp. 158–158.
- [32] Du Y., Ives R., Etter D. M., and Welch T., “A new approach to iris pattern recognition,” in *Electro-Optical and Infrared Systems: Technology and Applications*, vol. 5612. SPIE, 2004, pp. 104–116.
- [33] Chou C.-T., Shih S.-W., Chen W.-S., Cheng V. W., and Chen D.-Y., “Non-orthogonal view iris recognition system,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, pp. 417–430, 2009.
- [34] Zuo J. and Schmid N. A., “On a methodology for robust segmentation of nonideal iris images,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 703–718, 2009.
- [35] Tan C.-W. and Kumar A., “Unified framework for automated iris segmentation using distantly acquired face images,” *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4068–4079, 2012.
- [36] Yang W. S., Park B.-W., Jung E. H., Jeon N. J., Kim Y. C., Lee D. U., Shin S. S., Seo J., Kim E. K., Noh J. H. *et al.*, “Iodide management in formamidinium-lead-halide-based perovskite layers for efficient solar cells,” *Science*, vol. 356, no. 6345, pp. 1376–1379, 2017.
- [37] Morgado B., Vieira-Martins R., Assafin M., Machado D., Camargo J., Sfair R., Malacarne M., Braga-Ribas F., Robert V., Bassallo T. *et al.*, “Approx–mutual

- approximations between the galilean moons: the 2016–2018 observational campaign,” *Monthly Notices of the Royal Astronomical Society*, vol. 482, no. 4, pp. 5190–5200, 2019.
- [38] Abed K., El Morsi A. K., Sayed M. M., El Shaib A., and Gad M., “Effect of waste cooking-oil biodiesel on performance and exhaust emissions of a diesel engine,” *Egyptian journal of petroleum*, vol. 27, no. 4, pp. 985–989, 2018.
- [39] Minaee S., Abdolrashidiy A., and Wang Y., “An experimental study of deep convolutional features for iris recognition,” in *2016 IEEE signal processing in medicine and biology symposium (SPMB)*. IEEE, 2016, pp. 1–6.
- [40] Nagi J., Ducatelle F., Di Caro G. A., Cireşan D., Meier U., Giusti A., Nagi F., Schmidhuber J., and Gambardella L. M., “Max-pooling convolutional neural networks for vision-based hand gesture recognition,” in *2011 IEEE international conference on signal and image processing applications (ICSIPA)*. IEEE, 2011, pp. 342–347.
- [41] Johnson R., “Can iris patterns be used to identify people,” *Los Alamos National Laboratory, CA, Chemical and Laser Sciences Division, Rep. LA-12331-PR*, 1991.
- [42] Bracewell R. N., *Two-dimensional imaging*. Prentice-Hall, Inc., 1995.
- [43] Kalaiselvi S., UniversityKaraikudi A., India T., Jothi R., India T., and Palanisamy V., “Biometric security with iris recognition techniques: A review,” *International Journal of Pure and Applied Mathematics*, vol. 118, no. 8, pp. 567–572, 2018.
- [44] Jain A. K., Ross A., and Prabhakar S., “An introduction to biometric recognition,” *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.
- [45] He Z., Tan T., Sun Z., and Qiu X., “Toward accurate and fast iris segmentation for iris biometrics,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1670–1684, 2008.

- [46] Deng G. and Cahill L., “An adaptive gaussian filter for noise reduction and edge detection,” in *1993 IEEE conference record nuclear science symposium and medical imaging conference*. IEEE, 1993, pp. 1615–1619.
- [47] Al-Waisy A. S., Qahwaji R., Ipson S., and Al-Fahdawi S., “A fast and accurate iris localization technique for healthcare security system,” in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 1028–1034.
- [48] Sun Y. and Hua Y., “Image preprocessing of iris recognition,” in *2018 IEEE 3rd International Conference on Integrated Circuits and Microsystems (ICICM)*. IEEE, 2018, pp. 401–405.
- [49] Masek L. *et al.*, “Recognition of human iris patterns for biometric identification,” 2003.
- [50] Ghali A. A., Jamel S., Mohamad K. M., Yakub N. A., and Deris M. M., “A review of iris recognition algorithms,” *JOIV: International Journal on Informatics Visualization*, vol. 1, no. 4-2, pp. 175–178, 2017.
- [51] Daugman J. G., “High confidence visual recognition of persons by a test of statistical independence,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.
- [52] Wildes R. P., Asmuth J. C., Hanna K. J., Hsu S. C., Kolczynski R. J., Matey J. R., and McBride S. E., “Automated, non-invasive iris recognition system and method,” Nov. 5 1996, uS Patent 5,572,596.
- [53] Milletari F., Navab N., and Ahmadi S.-A., “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 fourth international conference on 3D vision (3DV)*. Ieee, 2016, pp. 565–571.
- [54] Ilhan A., Sekeroglu B., and Abiyev R., “Brain tumor segmentation in mri images using nonparametric localization and enhancement methods with u-net,”

International Journal of Computer Assisted Radiology and Surgery, vol. 17, no. 3, pp. 589–600, 2022.

- [55] Tran S.-T., Cheng C.-H., and Liu D.-G., “A multiple layer u-net, u n-net, for liver and liver tumor segmentation in ct,” *IEEE Access*, vol. 9, pp. 3752–3764, 2020.
- [56] Ronneberger O., Fischer P., and Brox T., “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
- [57] Kist A. M. and Döllinger M., “Efficient biomedical image segmentation on edgetpus at point of care,” *IEEE Access*, vol. 8, pp. 139 356–139 366, 2020.
- [58] Masek L. *et al.*, “Recognition of human iris patterns for biometric identification,” 2003.
- [59] Ghali A. A., Jamel S., Mohamad K. M., Yakub N. A., and Deris M. M., “A review of iris recognition algorithms,” *JOIV: International Journal on Informatics Visualization*, vol. 1, no. 4-2, pp. 175–178, 2017.
- [60] Minaee S., Abdolrashidiy A., and Wang Y., “An experimental study of deep convolutional features for iris recognition,” in *2016 IEEE signal processing in medicine and biology symposium (SPMB)*. IEEE, 2016, pp. 1–6.
- [61] Nguyen K., Fookes C., Ross A., and Sridharan S., “Iris recognition with off-the-shelf cnn features: A deep learning perspective,” *IEEE Access*, vol. 6, pp. 18 848–18 855, 2017.
- [62] Dalal N. and Triggs B., “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.
- [63] Simonyan K. and Zisserman A., “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

- [64] Xiao J., Wang J., Cao S., and Li B., “Application of a novel and improved vgg-19 network in the detection of workers wearing masks,” in *Journal of Physics: Conference Series*, vol. 1518, no. 1. IOP Publishing, 2020, p. 012041.
- [65] Lin Z., Niu Y. F., Rui Z. W., and Ling Z. X., “Review of full convolutional neural networks [j/ol],” *Computer Engineering and Applications*, pp. 1–14, 2019.
- [66] Khan M. S. M., Ahmed M., Rasel R. Z., and Khan M. M., “Cataract detection using convolutional neural network with vgg-19 model,” in *2021 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2021, pp. 0209–0212.
- [67] Boyd A., Czajka A., and Bowyer K., “Deep learning-based feature extraction in iris recognition: Use existing models, fine-tune or train from scratch?” in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2019, pp. 1–9.
- [68] Al-Waisy A. S., Qahwaji R., Ipson S., Al-Fahdawi S., and Nagem T. A., “A multi-biometric iris recognition system based on a deep learning approach,” *Pattern Analysis and Applications*, vol. 21, pp. 783–802, 2018.
- [69] Rosten E., Porter R., and Drummond T., “Faster and better: A machine learning approach to corner detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2008.
- [70] Kerl C., Sturm J., and Cremers D., “Dense visual slam for rgb-d cameras,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.
- [71] Youmaran R., *Algorithms to process and measure biometric information content in low quality face and iris images*. University of Ottawa (Canada), 2011.
- [72] Zhu R., Yang J., and Wu R., “Iris recognition based on local feature point matching,” in *2006 International Symposium on Communications and Information Technologies*. IEEE, 2006, pp. 451–454.

- [73] Mordini A., Peruzzi D., Russo F., Valacchi M., Reginato G., and Brandi A., "Superbase-promoted rearrangement of oxiranes to cyclopropanes," *Tetrahedron*, vol. 61, no. 13, pp. 3349–3360, 2005.
- [74] Kumar S. and Walia E., "Analysis of various biometric techniques," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 4, pp. 1595–1597, 2011.
- [75] Unar J., Seng W. C., and Abbasi A., "A review of biometric technology along with trends and prospects," *Pattern recognition*, vol. 47, no. 8, pp. 2673–2688, 2014.
- [76] Phillips P. J., Martin A., Wilson C. L., and Przybocki M., "An introduction evaluating biometric systems," *Computer*, vol. 33, no. 2, pp. 56–63, 2000.
- [77] Zebbiche K., Khelifi F., and Bouridane A., "An efficient watermarking technique for the protection of fingerprint images," *EURASIP journal on information security*, vol. 2008, pp. 1–20, 2008.
- [78] Chandran G. C. and Rajesh R., "Performance analysis of multimodal biometric system authentication," *Int. J. Comput. Sci. Network Security*, vol. 9, no. 3, 2009.
- [79] Scherhag U., Raghavendra R., Raja K. B., Gomez-Barrero M., Rathgeb C., and Busch C., "On the vulnerability of face recognition systems towards morphed face attacks," in *2017 5th international workshop on biometrics and forensics (IWBF)*. IEEE, 2017, pp. 1–6.
- [80] Nyah N., Jakaite L., Schetinin V., Sant P., and Aggoun A., "Learning polynomial neural networks of a near-optimal connectivity for detecting abnormal patterns in biometric data," in *2016 SAI computing conference (SAI)*. IEEE, 2016, pp. 409–413.
- [81] Oyallon E. and Rabin J., "An analysis of the surf method," *Image Processing On Line*, vol. 5, pp. 176–218, 2015.

- [82] Bücker M., “Alan julian izenman: Modern statistical techniques: regression, classification, and manifold learning,” *Statistical Papers*, vol. 53, no. 4, p. 1065, 2012.
- [83] Herbert B., “Luc van gool, surf: Speeded-up robust features,” in *Proceedings of the 9th European Conference on Computer Vision, 2006*, 2006.
- [84] Prabhakar S., *Fingerprint classification and matching using a filterbank*. Michigan State University, 2001.
- [85] Al Smadi T., Al Issa H. A., Trad E., Al Smadi K. A. *et al.*, “Artificial intelligence for speech recognition based on neural networks,” *Journal of Signal and Information Processing*, vol. 6, no. 02, p. 66, 2015.
- [86] Kaur G., Srivastava M., and Kumar A., “Speaker and speech recognition using deep neural network,” *International Journal of Emerging Research in Management and Technology*, vol. 6, p. 8, 2017.
- [87] Papastratis I., “Speech recognition: a review of the different deep learning approaches,” *Accessed on*, vol. 2, p. 2021, 2021.
- [88] Ross A., Jain A., and Pankati S., “A prototype hand geometry-based verification system,” in *Proceedings of 2nd conference on audio and video based biometric person authentication*, 1999, pp. 166–171.
- [89] Chiroma H., “Deep learning algorithms based fingerprint authentication: systematic literature review,” *Journal of Artificial Intelligence and Systems*, vol. 3, no. 1, pp. 157–197, 2021.
- [90] Maltoni D., Maio D., Jain A. K., Prabhakar S. *et al.*, *Handbook of fingerprint recognition*. Springer, 2009, vol. 2.
- [91] Liu M., “Fingerprint classification based on adaboost learning from singularity features,” *Pattern Recognition*, vol. 43, no. 3, pp. 1062–1070, 2010.
- [92] Ahsan M., Based M., Haider J., Kowalski M. *et al.*, “An intelligent system for automatic fingerprint identification using feature fusion by gabor filter and deep learning,” *Computers and Electrical Engineering*, vol. 95, p. 107387, 2021.

- [93] Cimtay Y., Alkan B., and Demirel B., “Fingerprint pattern classification by using various pre-trained deep neural networks,” *Avrupa Bilim ve Teknoloji Dergisi*, no. 24, pp. 258–261, 2021.
- [94] Adler F. H., “Physiology of the eye: clinical application,” (*No Title*), 1965.
- [95] Daugman J., “How iris recognition works,” in *The essential guide to image processing*. Elsevier, 2009, pp. 715–739.
- [96] Liu M., Zhou Z., Shang P., and Xu D., “Fuzzified image enhancement for deep learning in iris recognition,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 1, pp. 92–99, 2019.
- [97] Bouzouina Y. and Hamami L., “Multimodal biometric: Iris and face recognition based on feature selection of iris with ga and scores level fusion with svm,” in *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*. IEEE, 2017, pp. 1–7.
- [98] Nguyen K., Fookes C., Ross A., and Sridharan S., “Iris recognition with off-the-shelf cnn features: A deep learning perspective,” *IEEE Access*, vol. 6, pp. 18 848–18 855, 2017.
- [99] Alay N. and Al-Baity H. H., “Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits,” *Sensors*, vol. 20, no. 19, p. 5523, 2020.
- [100] Kim W., Song J. M., and Park K. R., “Multimodal biometric recognition based on convolutional neural network by the fusion of finger-vein and finger shape using near-infrared (nir) camera sensor,” *Sensors*, vol. 18, no. 7, p. 2296, 2018.
- [101] Boucherit I., Zmirli M. O., Hentabli H., and Rosdi B. A., “Finger vein identification using deeply-fused convolutional neural network,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 646–656, 2022.

- [102] Mróz-Gorgoń B., Wodo W., Andrych A., Caban-Piaskowska K., and Kozyra C., “Biometrics innovation and payment sector perception,” *Sustainability*, vol. 14, no. 15, p. 9424, 2022.
- [103] Mohammed R. S. and Ahmed A. A., “Iris recognition technology: Principles, mechanism, and market forecasting (2022-2030),” *PROCEEDINGS BOOK*, vol. 1, 2022.
- [104] Al-Waisy A. S., Qahwaji R., Ipson S., Al-Fahdawi S., and Nagem T. A., “A multi-biometric iris recognition system based on a deep learning approach,” *Pattern Analysis and Applications*, vol. 21, pp. 783–802, 2018.
- [105] Schuckers S. A., Schmid N. A., Abhyankar A., Dorairaj V., Boyce C. K., and Hornak L. A., “On techniques for angle compensation in nonideal iris recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1176–1190, 2007.
- [106] Ryan W. J., Woodard D. L., Duchowski A. T., and Birchfield S. T., “Adapting starburst for elliptical iris segmentation,” in *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*. IEEE, 2008, pp. 1–7.
- [107] Pundlik S. J., Woodard D. L., and Birchfield S. T., “Non-ideal iris segmentation using graph cuts,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2008, pp. 1–6.
- [108] Lozej J., Štepec D., Štruc V., and Peer P., “Influence of segmentation on deep iris recognition performance,” in *2019 7th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2019, pp. 1–6.
- [109] Chen L.-C., Zhu Y., Papandreou G., Schroff F., and Adam H., “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

- [110] Howard A., Zhmoginov A., Chen L.-C., Sandler M., and Zhu M., “Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation,” 2018.
- [111] Chollet F., “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [112] Daugman J., “New methods in iris recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1167–1175, 2007.
- [113] Wang C., Zhu Y., Liu Y., He R., and Sun Z., “Joint iris segmentation and localization using deep multi-task learning framework,” *arXiv preprint arXiv:1901.11195*, 2019.
- [114] Simonyan K. and Zisserman A., “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [115] Huang G., Liu S., Van der Maaten L., and Weinberger K. Q., “Condensenet: An efficient densenet using learned group convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2752–2761.
- [116] Zhao Z. and Kumar A., “Towards more accurate iris recognition using deeply learned spatially corresponding features,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3809–3818.
- [117] Simonyan K. and Zisserman A., “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [118] Zhao Z. and Kumar A., “Towards more accurate iris recognition using deeply learned spatially corresponding features,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3809–3818.
- [119] Long J., Shelhamer E., and Darrell T., “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

- [120] Umer S., Dhara B. C., and Chanda B., “A novel cancelable iris recognition system based on feature learning techniques,” *Information Sciences*, vol. 406, pp. 102–118, 2017.
- [121] Jain A. K., Ross A., and Prabhakar S., “An introduction to biometric recognition,” *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.
- [122] Cireşan D. C., Meier U., and Schmidhuber J., “Transfer learning for latin and chinese characters with deep neural networks,” in *The 2012 international joint conference on neural networks (IJCNN)*. IEEE, 2012, pp. 1–6.
- [123] Li L., Abd-El-Atty B., Elseuofi S., Abd El-Rahiem B., and Abd El-Latif A. A., “Quaternion and multiple chaotic systems based pseudo-random number generator,” in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–5.
- [124] Song C., Huang Y., Wang W., and Wang L., “Casia-e: a large comprehensive dataset for gait recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2801–2815, 2022.
- [125] Andrie R., Basuki A., and Arai K., “A review of chinese academy of sciences (casia) gait database as a human gait recognition dataset,” 2011.
- [126] Umer S., Dhara B. C., and Chanda B., “Iris recognition using multiscale morphologic features,” *Pattern Recognition Letters*, vol. 65, pp. 67–74, 2015.
- [127] Moran F., Natarajan S., and Nikolopoulou M., “Developing a database of energy use for historic dwellings in bath, uk,” *Energy and Buildings*, vol. 55, pp. 218–226, 2012.
- [128] Ho C. C., Ng H., Tan W.-H., Ng K.-W., Tong H.-L., Yap T. T.-V., Chong P.-F., Eswaran C., and Abdullah J., “Mmu gaspfa: a cots multimodal biometric database,” *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2043–2050, 2013.

- [129] Proença H. and Alexandre L. A., “The nice. i: noisy iris challenge evaluation-part i,” in *2007 First IEEE international conference on biometrics: theory, applications, and systems*. IEEE, 2007, pp. 1–4.
- [130] Clark N. N., Prucz J. C., Gautam M., Guatam M., and Lyons D. W., “The west virginia university heavy duty vehicle emissions database as a resource for inventory and comparative studies,” *SAE transactions*, pp. 2587–2604, 2000.
- [131] Parashar R. and Joshi S., “Comparative study of iris databases and ubiris database for iris recognition methods for non-cooperative environment,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, no. 5, 2012.
- [132] Nguyen D.-L., Putro M. D., and Jo K.-H., “Human eye detector with light-weight and efficient convolutional neural network,” in *Advances in Computational Collective Intelligence: 12th International Conference, ICCCI 2020, Da Nang, Vietnam, November 30–December 3, 2020, Proceedings 12*. Springer, 2020, pp. 186–196.
- [133] He Z., Tan T., Sun Z., and Qiu X., “Toward accurate and fast iris segmentation for iris biometrics,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1670–1684, 2008.
- [134] Choi J. H., Lee K. I., Kim Y. C., and Song B. C., “Accurate eye pupil localization using heterogeneous cnn models,” in *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019, pp. 2179–2183.
- [135] Zhang X., Sugano Y., Fritz M., and Bulling A., “Appearance-based gaze estimation in the wild,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4511–4520.
- [136] Stojmenovic M. and Nayak A., “Direct ellipse fitting and measuring based on shape boundaries,” in *Advances in Image and Video Technology: Second Pacific Rim Symposium, PSIVT 2007 Santiago, Chile, December 17-19, 2007 Proceedings 2*. Springer, 2007, pp. 221–235.

- [137] Hassanein A. S., Mohammad S., Sameer M., and Ragab M. E., “A survey on hough transform, theory, techniques and applications,” *arXiv preprint arXiv:1502.02160*, 2015.
- [138] Masters D. and Luschi C., “Revisiting small batch training for deep neural networks,” *arXiv preprint arXiv:1804.07612*, 2018.
- [139] Ioffe S. and Szegedy C., “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. pmlr, 2015, pp. 448–456.
- [140] Huang G., Liu Z., Van Der Maaten L., and Weinberger K. Q., “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [141] He K., Zhang X., Ren S., and Sun J., “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [142] Santurkar S., Tsipras D., Ilyas A., and Madry A., “How does batch normalization help optimization?” *Advances in neural information processing systems*, vol. 31, 2018.
- [143] Meng X., Cai L., Xu X., Yang X., Shen X., Dong G., and Wang Y., “Two-step phase-shifting interferometry and its application in image encryption,” *Optics letters*, vol. 31, no. 10, pp. 1414–1416, 2006.
- [144] Zhang X., Wu X. Q., Lu S., Guo Y. L., and Ma X., “Deficit of mitochondria-derived atp during oxidative stress impairs mouse mii oocyte spindles,” *Cell research*, vol. 16, no. 10, pp. 841–850, 2006.
- [145] Felzenszwalb P. F., Girshick R. B., McAllester D., and Ramanan D., “Object detection with discriminatively trained part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

- [146] Zhou W., Gao S., Zhang L., and Lou X., “Histogram of oriented gradients feature extraction from raw bayer pattern images,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 946–950, 2020.
- [147] Gonzalez R. C. and Wintz P., “Digital image processing(book),” *Reading, Mass., Addison-Wesley Publishing Co., Inc.(Applied Mathematics and Computation*, no. 13, p. 451, 1977.
- [148] Singh A., Pandey A., Rakhra M., Singh D., Singh G., and Dahiya O., “An iris recognition system using cnn & vgg16 technique,” in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2022, pp. 1–6.
- [149] Khan M. S. M., Ahmed M., Rasel R. Z., and Khan M. M., “Cataract detection using convolutional neural network with vgg-19 model,” in *2021 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2021, pp. 0209–0212.
- [150] Boyd A., Czajka A., and Bowyer K., “Deep learning-based feature extraction in iris recognition: Use existing models, fine-tune or train from scratch?” in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2019, pp. 1–9.
- [151] Yosinski J., Clune J., Nguyen A., Fuchs T., and Lipson H., “Understanding neural networks through deep visualization,” *arXiv preprint arXiv:1506.06579*, 2015.
- [152] Li Y. and Yuan Y., “Convergence analysis of two-layer neural networks with relu activation,” *Advances in neural information processing systems*, vol. 30, 2017.
- [153] Tobji R., Di W., Ayoub N., and Haouassi S., “Efficient iris pattern recognition method by using adaptive hamming distance and 1-d log-gabor filter,” *Int. J. Adv. Comput. Sci. Appl*, vol. 9, no. 11, pp. 662–669, 2018.
- [154] Sultana F., Sufian A., and Dutta P., “Advancements in image classification using convolutional neural network,” in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2018, pp. 122–129.

- [155] Jabbar H. and Khan R. Z., “Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study),” *Computer Science, Communication and Instrumentation Devices*, vol. 70, no. 10.3850, pp. 978–981, 2015.
- [156] Deng J., Dong W., Socher R., Li L.-J., Li K., and Fei-Fei L., “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [157] Yosinski J., Clune J., Nguyen A., Fuchs T., and Lipson H., “Understanding neural networks through deep visualization,” *arXiv preprint arXiv:1506.06579*, 2015.
- [158] Chaudhari R. D., Pawar A. A., and Deore R. S., “The historical development of biometric authentication techniques: A recent overview,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 10, 2013.
- [159] Paderes R. E. O., “A comparative review of biometric security systems,” in *2015 8th International Conference on Bio-Science and Bio-Technology (BSBT)*. IEEE, 2015, pp. 8–11.
- [160] Kong W.-K. and Zhang D., “Detecting eyelash and reflection for accurate iris segmentation,” *International journal of pattern recognition and artificial intelligence*, vol. 17, no. 06, pp. 1025–1034, 2003.
- [161] Ma L., Tan T., Wang Y., and Zhang D., “Efficient iris recognition by characterizing key local variations,” *IEEE Transactions on Image processing*, vol. 13, no. 6, pp. 739–750, 2004.
- [162] Nowak R. D. and Baraniuk R. G., “Adaptive weighted highpass filters using multiscale analysis,” *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1068–1074, 1998.
- [163] Mallat S. G., “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

- [164] Wildes R. P., Asmuth J. C., Hanna K. J., Hsu S. C., Kolczynski R. J., Matey J. R., and McBride S. E., “Automated, non-invasive iris recognition system and method,” Nov. 5 1996, uS Patent 5,572,596.
- [165] Kolmogorov V. and Zabini R., “What energy functions can be minimized via graph cuts?” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 2, pp. 147–159, 2004.
- [166] Boykov Y., Kolmogorov V., Cremers D., and Delong A., “An integral solution to surface evolution pdes via geo-cuts,” in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III* 9. Springer, 2006, pp. 409–422.
- [167] Boykov Y. and Funka-Lea G., “Graph cuts and efficient nd image segmentation,” *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.
- [168] Pundlik S. J., Woodard D. L., and Birchfield S. T., “Non-ideal iris segmentation using graph cuts,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2008, pp. 1–6.
- [169] Daugman J., “How iris works|| ieee transaction on circuit and systems for video technology,” 2004.
- [170] Zhou Z., Du Y., and Belcher C., “Transforming traditional iris recognition systems to work in nonideal situations,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, pp. 3203–3213, 2009.

APPENDICES

Appendix A: Programming Language and Libraries

Appendix B: Software Code

Appendix C: Biometric Method.



Appendix A – Programming Language and Libraries

This appendix contains the programming language, libraries, methods, and tools, it is allocated to explain all tools I have used in this research, define the necessary and required libraries, and clarify how to use them through the iris identification phases, the usage of Python, python libraries, deep learning, computer vision and transfer learning in this research. Python programming language, an interpreted, object-oriented, high-level programming language with dynamic semantics, has been used as a base programming language in this research.

This research is based on deep learning as defined in the title, Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans, as demonstrated in Figure A.1 Deep learning is a subfield of machine learning and artificial intelligence, extremely beneficial to data scientists, to handle growing volume and availability of data, the backbone to develop autonomous systems and has the ability to completely use large amounts of data.

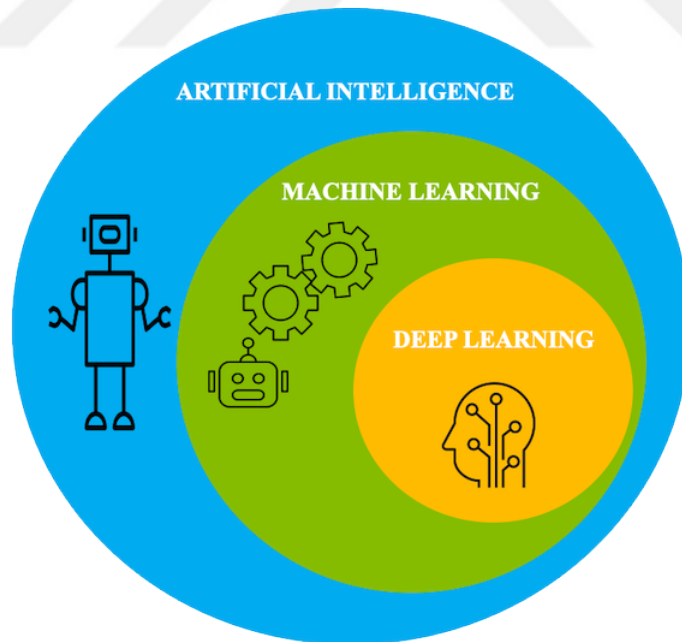


Figure A.1 DL, ML, and AI concept [IBM Developer]

Lately, deep learning has made a considerable influence and generated excellent results in biometrics systems [1]. Deep learning algorithms have dominated a large number of the limitations of other machine learning algorithms, principally those

related to feature extraction techniques. Deep learning algorithms can cope with biometric image transformations and extract features from raw data [2]. Machine learning algorithms need almost extraction techniques to extract features from raw biometric data and transform the raw data into a suitable and reliable format before classifying it [3]. Furthermore, machine learning algorithms necessitate and involve certain preprocessing stages to be derived prior to feature extraction. Additionally, some extraction techniques do not continuously work positively with different types of biometrics or different data sets of the same biometrics.

I have used some Python libraries such as OpenCV in order to perform image processing and computer vision tasks. In this research, it provides a wide range of characteristics, such as image processing, image filtering, face detection, eye detection, and tracking and detection. OpenCV has been used in OpenCV in Gaussian blurring, also in it is used in the mean filter and in all parts of the image preprocessing and edge detection.

The next library to talk about is NumPy Which stands for Numerical Python and is the main library for numeric and scientific computing. NumPy contains multidimensional array substances and a collection of routines for processing those arrays. It is a library for the Python programming language, in addition, it supports large, multi-dimensional matrices and arrays, laterally with a large collection of high-level mathematical functions to work on these arrays. In this research, I have used NumPy in image filtering such as mean filter, Ellipse Fitting, histogram equalization, exponential operator, Gaussian filter, and edge detection.

Python Matplotlib which is a plotting library for the Python programming language and its numerical mathematics extension NumPy is also used in this research and plays an important role in all parts of the iris identification. It affords an object-oriented API for implanting plots into applications using general-purpose GUI toolkits. Matplotlib is a low-level graph plotting library in Python programming language that obliges as a visualization utility, in this research work I have used Matplotlib to visualize the iris images, the filters, and edge detection, loss, and accuracy results.

Appendix B – Software Code

This appendix includes the coding part and dedicated to explaining the program and software code I have been used in this research, defining the essential and required data, and clarifying how to achieve the result through the iris identification phases, the usage of deep learning, computer vision and transfer learning in this study. Python programming language which is an interpreted, object-oriented, high-level programming language with dynamic semantics has been used in order to develop the software and to preprocess the iris images, which is the basis for image processing to obtain desired accuracy in image segmentation, feature extraction, and image classification. The following sections explain the iris identification software code phases.

Image preprocessing Software Code

To attain this purpose, image filtering and edge detection are needed for iris image preprocessing. Gaussian filtering has been effectively and exhaustively has been studied in image processing and computer vision. Image filtering is the first step in iris image preprocessing, which is to omit the intrusion and interference in the original image for edge detection. in this thesis, the Gaussian filter has been used to reduce the search space the eye image needs to be enhanced before and after generating the edge map. In the first step of all sections, the most important and essential part is importing the iris image dataset. I imported the dataset from my Google Drive and here I explain the functions which are used to load the images and the data preparation. Libraries used in image preprocessing code are illustrated in Figure B.1.

In any deep learning project libraries installation, and importing the libraries are the first and the most important and obligatory steps. In this step, I imported Python and Deep Learning libraries specifically: TensorFlow, Keras, and matplotlib which make my work facile.

```
[ ] import matplotlib as plt
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
from matplotlib.image import imread
import seaborn as sns
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dropout, Flatten, Dense
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
%matplotlib inline
```

Figure B.1 Libraries used in image preprocessing

Downloading and uploading the databases could be the first step in image data preprocessing, and I connected my Google Drive to my Google Collab through the following code in Figure B.2:

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Figure B.2 Importing the Google Drive

Iris images are in a zip file in order to extract them the following code in Figure B.3 has been used:

```
[ ] # Extracting the compressed Dataset
import zipfile

from zipfile import ZipFile
Iris_Images = '/content/drive/MyDrive/Iris Database/Iris_Images.zip'
with ZipFile(Iris_Images, 'r') as zip:
    zip.extractall()
    print('The database is extracted')
```

The database is extracted

Figure B.3 Extracting the compressed Dataset code

Figure B.4 Visualize one image and draw eye corners and iris center, which is very important in image processing, especially in this study to detect the eyes and iris center, in addition, if the full-face image is given in the initial step, it is significant to detect the eye. Then I used the code in Figure B.5 to crop the eye from the full-face image to make the process simple.

```
[ ] Upper_Left = (x1[12]+7, y1[12]-20)
    Bottom_Right = (x3[12]-7, y3[12]+20)
    L_center_coordinates = (x2[12], y2[12])
    Upper_Right = (x4[12]+7, y4[12]+20)
    Bottom_Left = (x6[12]-7, y6[12]-20)
    R_center_coordinates = (x5[12], y5[12])

    eye_detect_image = face_image[12].copy()
    eye_detect_image = cv2.rectangle(eye_detect_image, Upper_Left, Bottom_Right,
                                     color=(0,255,0), thickness=1)
    eye_detect_image = cv2.circle(eye_detect_image, L_center_coordinates, radius=0,
                                  color=(255,0,0), thickness=2)
    eye_detect_image = cv2.rectangle(eye_detect_image, Upper_Right, Bottom_Left,
                                     color=(0,255,0), thickness=1)
    eye_detect_image = cv2.circle(eye_detect_image, R_center_coordinates, radius=0,
                                  color=(255,0,0), thickness=2)

    cv2.imshow(eye_detect_image)
    print(eye_detect_image.shape)
```

Figure B.4 Detecting the eye and iris center code

```
[ ] crop_images = []
    x_iris = np.zeros(x2.shape,dtype=int)
    y_iris = np.zeros(y2.shape,dtype=int)
    for j in range(len(images)):
        mx = int((x1[j] + x3[j])/2)
        my = int((y3[j] + y1[j])/2)

        Xc = mx-24
        Yc = my-24

        x_iris[j] = x2[j]-Xc
        y_iris[j] = y2[j]-Yc

        crop_images.append(images[j][Yc:Yc+48 , Xc:Xc+48])

    plt.axis("off")
    plt.imshow(crop_images[12], cmap='gray')
    plt.show()
    print(crop_images[12].shape)
```

Figure B.5 Cropping the eye image from the full-face image

After cropping the eye images, I normalized the images in order to ensure that each image comes from a standard distribution, this could be because of the range of pixel values in one input image and also the same as the range in other images. Standardization aims to make the model train and reach a minimum error, satisfying accuracy, faster. The pixel values naturally range from 0 to 255, which is why before feeding images into a convolutional neural network, it is a good idea to normalize them in some way so as to put all pixels in a manageable range. Figure B.6 shows the cropped image after normalization.

```
[ ] crop_images = np.array(crop_images)
    iris_images = crop_images/255.0

    plt.axis("off")
    plt.imshow(iris_images[12], cmap='gray')
    plt.show()
```

Figure B.6 Normalizing the cropped eye images code

To increase the accuracy, and reduce the error rate applying Dilation and Gaussian filtering is an essential issue after applying the normalization. Filtering does not satisfy and end here, I will discuss more about it in the edge detection and filtering section. In Figure B.7, I Created final datasets X, and Y, and Set the kernel for dilation.

```
[ ] from scipy.ndimage import gaussian_filter

    X = iris_images
    Y = np.zeros(X.shape)

    #Set kernel for dilation
    kernel = np.ones((3,3),np.uint8)

    for k in range(len(X)):
        img = np.zeros((48, 48))
        img[x_iris[k],y_iris[k]] = 1

        img = gaussian_filter(img, 1)
        img = cv2.dilate(img,kernel,iterations = 1)

        Y[k] = img
```

Figure B.7 Applying Dilation and Gaussian filtering code

The train-test split process is beneficial to estimate the efficiency and performance of deep learning algorithms, I used it to make predictions on the data. When splitting the dataset it could be necessary for unbiased evaluation of prediction performance. The test is supposed as 20% of the entire dataset. X, Y are expanded by 1 dimension to match the dimensions of the CNN model. Figure B.8 demonstrates the train-test split code in deep learning.

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)

    X_train = np.expand_dims(X_train, 1)
    Y_train = np.expand_dims(Y_train, 1)
    X_test = np.expand_dims(X_test, 1)
    Y_test = np.expand_dims(Y_test, 1)

    print(f'X_test size {X_test.shape} , Y_test size {Y_test.shape}')

X_test size (248, 1, 48, 48) , Y_test size (248, 1, 48, 48)
```

Figure B.8 splitting data into train and test set code

After splitting the images into X_train, Y_train, X_test, and Y_test image data augmentation was applied. Image data augmentation is a method used that obtain new images from existing Images. In order to do augmentation, I bring some tiny changes to them, in other words, these changes include rotating the images, adjusting the brightness of the images, shifting the subject in the image horizontally or vertically, and zooming the images in case of need. Figure Figure B.9 demonstrates the image data augmentation code.

```
[ ] traindata_gen=ImageDataGenerator(
    rotation_range=0,
    rescale=1/255.,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    fill_mode='nearest'
)

testdata_gen=ImageDataGenerator(

    rescale=1./255)

[ ] traindata_gen.flow_from_directory(dataset_path)
```

Figure B.9 Image data augmentation code

After Image data augmentation, Autoencoder is applied which contains the Base network, Auxiliary Network, and Skip Connection was also applied. An autoencoder is used to reduce the image noise or coloring. Figure B.10 demonstrates the autoencoder class and the forward method code.

```
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()

        #Encoder layers:
        self.conv1 = nn.Conv2d(1, 64, 3, padding=1)
        self.conv2 = nn.Conv2d(64, 128, 3, padding=1)
        self.pool1 = nn.MaxPool2d(2, 2)
        self.conv3 = nn.Conv2d(128, 256, 3, padding=1)
        self.pool2 = nn.MaxPool2d(2, 2)

        #Base Network:
        self.conv1_T = nn.ConvTranspose2d(256, 64, 4, stride=4)
        self.conv4 = nn.Conv2d(128, 64, 3, padding=1)
        self.conv5 = nn.Conv2d(64, 1, 3, padding=1)

        #Auxiliary Network:
        self.conv2_T = nn.ConvTranspose2d(256, 128, 4, stride=2, padding=1)
        self.conv3_T = nn.ConvTranspose2d(128, 64, 4, stride=2, padding=1)

    def forward(self, x):
        #Encoder:
        x_skip = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x_skip))
        x = self.pool1(x)
        x = F.relu(self.conv3(x))
        x = self.pool2(x)

        #Base Network:
        y = F.relu(self.conv1_T(x))
        y = torch.cat((x_skip, y), 1)
        y = F.relu(self.conv4(y))
        y = torch.tanh(self.conv5(y))

        #Auxiliary Network:
        ax = F.relu(self.conv2_T(x))
        ax = F.relu(self.conv3_T(ax))
        ax = torch.tanh(self.conv5(ax))
```

Figure B.10 Autoencoder Class and the forward method code

Then in Figure B.11, I created a CNN model to detect the iris center. In this model, the X and Y are transformed into torch tensors then the data loaders are built after that model training and testing are performed. After running the model with different

optimizers : [sgd, adam, adamax, rmsprop], loss functions : [mse,mae], and activations : [tanh, relu, sigmoid], and Optimizer = Adam, Learning rate = 0.001 The number of epochs = 100, the Loss function is MSE and the Activation function is Tanh. The code in Figure B.12 visualizes the predicted iris centers.

```
] tensor_x_tr = torch.Tensor(X_train)
   tensor_y_tr = torch.Tensor(Y_train)
   tensor_x_test = torch.Tensor(X_test)
   tensor_y_test = torch.Tensor(Y_test)

   epochs = 100
   learning_rate = 0.001
   batchSize = 64
   criterion = nn.MSELoss()
   optimizer = torch.optim.Adam(model.parameters(),lr=learning_rate)

   train_loader = DataLoader(TensorDataset(tensor_x_tr,tensor_y_tr)
                             ,batch_size=batchSize,shuffle=True)
   test_loader = DataLoader(TensorDataset(tensor_x_test,tensor_y_test)
                             ,batch_size=batchSize, shuffle=False)

   for epoch in range(1, epochs + 1):
       train(model, device, train_loader, optimizer, epoch, criterion)
       iris_pred , iris_img = test(model, device, test_loader, criterion)
       print(epoch)
```

Figure B.11 CNN model to detect the iris center code

```
] fig = plt.figure(figsize=[4,20])
   for i in range(0,10):
       y_iris_pred = TesnorToImage(iris_pred[i+3])
       view_img = TesnorToImage(iris_img[i+3])

       (minVal, maxVal, minLoc, maxLoc) = cv2.minMaxLoc(y_iris_pred)

       fig.add_subplot(10,2,i*2+1)
       plt.imshow(y_iris_pred,cmap='gray')

       fig.add_subplot(10,2,i*2+2)
       plt.imshow(view_img,cmap='gray')
       plt.plot(maxLoc[0], maxLoc[1], 'b.', markersize=20)
```

Figure B.12 Visualizing the predicted iris centers using matplotlib

Image filtering is the most important part of iris image preprocessing, which is to omit the intrusion and interference in the original image for edge detection. The important goal of an image preprocessing filter involves Noise Reduction, Color Normalization, Edge Detection, and Histogram Equalization, and iris image preprocessing filters are

primarily used to support the high frequencies in the image or the low frequencies in the image. With effective image filtering, Iris detection will be guaranteed. Before implementing the Mean Filter and Gaussian Filter I implemented an exponential operator. Which is given an exponent r computed for each input pixel P_{in} the output pixel value P_{out} Figure B.13 explains the code.

```
def exponential_operator(img, r):
    exp_img = ((img/255)**r) *255
    return exp_img

# Extracting the histogram
hist,bins = np.histogram(img.flatten(),256,[0,256])

# Plotting the histogram before operator
plt.title("Hist Before Stretching")
plt.plot(hist, color = 'b')
plt.show()

exp_img = exponential_operator(img, 0.45)
hist_st,bins = np.histogram(exp_img.astype(np.uint8).flatten(),256,[0,256])
plt.imshow(exp_img,cmap='gray',vmin=0,vmax=255)
plt.show()
plt.title("Hist After Operator")
plt.plot(hist_st, color = 'b')
plt.show()
```

Figure B.13 Code which is Extracting and plotting the histogram

The Gaussian filter is used to blur the required region of an image and dispel the noise with superior frequencies. Its performance is the same as the mean filter and is the linear filter, though illustrating the average weight uniformly, which blurs the edge effectively and reduces the noise. I applied a Gaussian filter with $\sigma = 1.15$ to an image, the code in Figure B.14 finding and applying the Gaussian kernel and Using the OpenCV function for Gaussian blurring. The code in Figure B.15 explains the Hough Transform.

After Filtering I implemented edge detection which is an image processing method to find the boundaries of substances through the images, it detects interruptions in brightness. Edge detection is commonly used for image segmentation and data extraction in areas such as image preprocessing, machine vision, and computer vision.

```

sigma = 1.5
k_size = int(np.ceil((3*sigma))*2 + 1)
print("Sigma: {}, Kernel Size: {}".format(sigma, k_size))

gaussian_kernel_1D = cv2.getGaussianKernel(k_size, sigma)

start = time.time()
img_tr1 = cv2.filter2D(img,-1,gaussian_kernel_1D)
img_tr = cv2.filter2D(img_tr1,-1,gaussian_kernel_1D.transpose())
print("Time needed twice 1D: ", time.time()-start)
plt.imshow(img_tr,cmap='gray',vmin=0,vmax=255)
plt.show()

gaussian_kernel_2D = gaussian_kernel_1D.dot(gaussian_kernel_1D.transpose())

start = time.time()
img_tr = cv2.filter2D(img,-1,gaussian_kernel_2D)
print("Time needed single 2D (SLOWER): ", time.time()-start)

plt.imshow(img_tr,cmap='gray',vmin=0,vmax=255)
plt.show()

start = time.time()
img_tr = cv2.GaussianBlur(img, (k_size,k_size) , sigma)
print("Time needed with OpenCV API (Best Optimization): ", time.time()-start)

```

Figure B.14 Gaussian filter code which blurs the required region of an image

```

[ ] if img.shape[-1] == 3:
    b,g,r = cv2.split(img)
    rgb_img = cv2.merge([r,g,b])
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
else:
    gray_img = img

img = cv2.medianBlur(gray_img, 5)
cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)

circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,180,
                           param1=100,param2=30,minRadius=0,maxRadius=0)

circles = np.uint16(np.around(circles))

for i in circles[0,:]:
    cv2.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2)
    cv2.circle(cimg,(i[0],i[1]),20,(0,0,255),3)

```

Figure B.15 Hough Transform detecting the outer and inner boundaries of iris

Figure B.16 shows the ellipse fitting iris edge detection. Ellipticity become famous in industry and nature, and the method of identifying is important in the identification system in the area of agricultural and medical imaging systems for identifying certain grains, onions, watermelons, cells, and even human faces as in this research, it is used for iris edge detection [136]

```
def fit_rotated_ellipse_ransac(data,iter=30,sample_num=10,offset=80.0):

    count_max = 0
    effective_sample = None

    for i in range(iter):
        sample = np.random.choice(len(data), sample_num, replace=False)

        xs = data[sample][:,0].reshape(-1,1)
        ys = data[sample][:,1].reshape(-1,1)

        J = np.mat( np.hstack((xs*ys,ys**2,xs,
                                ys, np.ones_like(xs,dtype=np.float))) )
        Y = np.mat(-1*xs**2)
        P= (J.T * J).I * J.T * Y

        # fitter  $a*x^2 + b*x*y + c*y^2 + d*x + e*y + f = 0$ 
        a = 1.0; b= P[0,0]; c= P[1,0]; d = P[2,0]; e= P[3,0]; f=P[4,0];
        ellipse_model = lambda x,y : a*x**2 + b*x*y + c*y**2 + d*x + e*y + f
```

Figure B.16 Ellipse Fitting Iris Edge Detection code

Image Segmentation Software Code

Figure Figure B.17 exhibits the architecture of the U-Net model in this research with an iris image size of 240×320 pixels as an input.

```
[ ] # Define U-net architecture
def unet_model(input_layer, start_neurons):
    # Contraction path
    conv1 = Conv2D(start_neurons, kernel_size=(3, 3), activation="relu", padding="same")(input_layer)
    conv1 = Conv2D(start_neurons, kernel_size=(3, 3), activation="relu", padding="same")(conv1)
    pool1 = MaxPooling2D((2, 2))(conv1)
    pool1 = Dropout(0.25)(pool1)

    conv2 = Conv2D(start_neurons*2, kernel_size=(3, 3), activation="relu", padding="same")(pool1)
    conv2 = Conv2D(start_neurons*2, kernel_size=(3, 3), activation="relu", padding="same")(conv2)
    pool2 = MaxPooling2D((2, 2))(conv2)
    pool2 = Dropout(0.5)(pool2)

    conv3 = Conv2D(start_neurons*4, kernel_size=(3, 3), activation="relu", padding="same")(pool2)
    conv3 = Conv2D(start_neurons*4, kernel_size=(3, 3), activation="relu", padding="same")(conv3)
    pool3 = MaxPooling2D((2, 2))(conv3)
    pool3 = Dropout(0.5)(pool3)

    conv4 = Conv2D(start_neurons*8, kernel_size=(3, 3), activation="relu", padding="same")(pool3)
    conv4 = Conv2D(start_neurons*8, kernel_size=(3, 3), activation="relu", padding="same")(conv4)
    pool4 = MaxPooling2D((2, 2))(conv4)
    pool4 = Dropout(0.5)(pool4)
```

Figure B.17 The U-Net architecture code with CNN

The model compilation is an activity performed after writing the statements in a model and before training starts. It checks for format errors and defines the loss function, the optimizer or learning rate, and the metrics [122]. A compiled model is needed for training. In Figure B.18 we can see the loss function is 'binary_crossentropy', the optimizer is 'adam' and the metrics are 'accuracy'.

```
[ ] # Compile unet model
input_layer = Input((SIZE, SIZE, 3))
output_layer = unet_model(input_layer, start_neurons = 16)

model = Model(input_layer, output_layer)
model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 128, 128, 3)]	0	[]
conv2d_19 (Conv2D)	(None, 128, 128, 16)	448	['input_2[0][0]']

Figure B.18 Compiling the U-Net model

In U-Net segmentation, before training the model I have defined a post Process method to obtain a convincing result, and a Callback class to show the progress of learning on the images after each epoch, a callback can be an object that is able to perform actions at various stages of training (e.g. at the end or start of an epoch, before or after a batch). The code is illustrated in Figure B.19.

```
[ ] # Post Process
def mask_threshold(image, threshold=0.3):
    return image>threshold

class ShowProgress(Callback):
    def __init__(self, save=False):
        self.save = save
    def on_epoch_end(self, epoch, logs=None):
        k = np.random.randint(len(images_train))
        original_image = images_train[k][np.newaxis,...]
        predicted_mask = self.model.predict(original_image).reshape(128,128)
        proc_mask02 = mask_threshold(predicted_mask, threshold=0.2)
        proc_mask03 = mask_threshold(predicted_mask, threshold=0.3)
        proc_mask04 = mask_threshold(predicted_mask, threshold=0.4)
        # proc_mask05 = mask_threshold(predicted_mask, threshold=0.5)
        mask = mask_train[k].reshape(128,128)

        plt.figure(figsize=(15,10))

        plt.subplot(1,7,1)
        plt.grid(False)
        plt.imshow(original_image[0]);
        plt.title('Original Image')

        plt.subplot(1,7,2)
        plt.grid(False)
        plt.imshow(predicted_mask, cmap='gray');
        plt.title('Predicted Mask: 0.1')
```

Figure B.19 Post Process & Callback class in iris image segmentation

The code in Figure B.20 explains the U-Net model training with 150 epochs and a batch

size of 32.

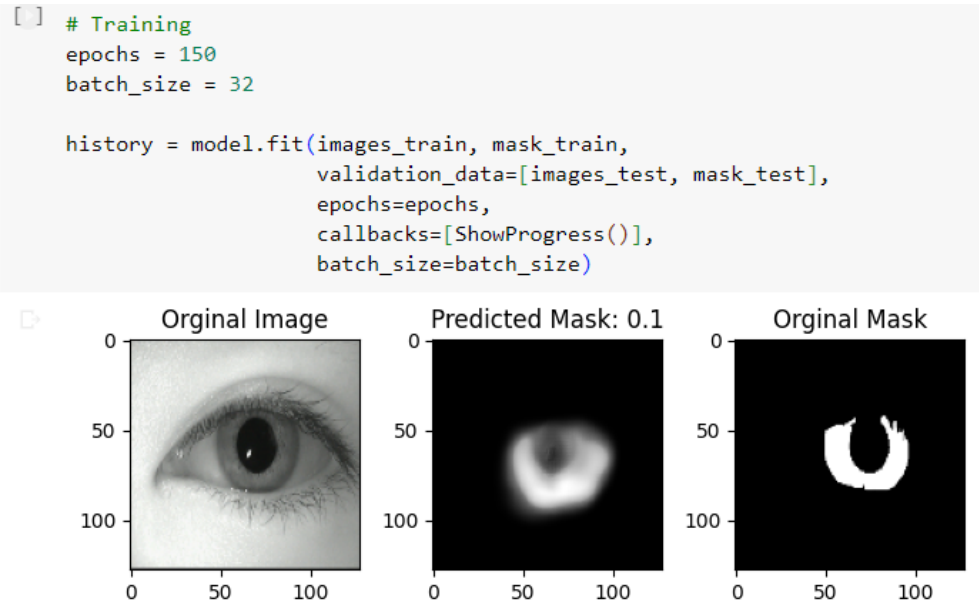


Figure B.20 U-Net Model Training code with 150 epochs and a batch size of 32

The encoder part consists of the frequentative convolutional layer which has a kernel size of 3×3 , in addition, each pursues a 2×2 max-pooling by step 2, which regularly decreases the batch size. The decoder part contains the up-sampling processes with a 2×2 convolution, which splits the number of feature channels in half.

```
[ ] plt.figure(figsize=(10,5))

# Plot loss for each epoch
plt.subplot(1,2,1)
plt.plot(history.history['loss'], label="Train loss")
plt.plot(history.history['val_loss'], label="Test loss")
plt.title('Training and Test Model Loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['TRAIN', 'TEST'], loc='upper left')

# Plot accuracy for each epoch
plt.subplot(1,2,2)
plt.plot(history.history['accuracy'], label="Train accuracy")
plt.plot(history.history['val_accuracy'], label="Test accuracy")
plt.title('Training and Test Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['TRAIN', 'TEST'], loc='upper left')

plt.tight_layout()
plt.show()
```

Figure B.21 Code to visualize the acc and loss for each epoch in segmentation

```
def plot_results(threshold=0.5):
    k = np.random.randint(len(images_test))
    original_image = images_test[k]
    mask = mask_test[k].reshape(128,128)
    predicted_mask = predictions[k].reshape(128,128)
    threshold_mask = mask_threshold(predicted_mask, threshold=threshold)

    plt.figure(figsize=(15,5))

    plt.subplot(1,4,1)
    plt.imshow(original_image);plt.title('Original Image')

    plt.subplot(1,4,2)
    plt.imshow(mask, cmap='gray');plt.title('Original Mask')

    plt.subplot(1,4,3)
    plt.imshow(predicted_mask, cmap='gray');plt.title('Predicted Mask')

    plt.subplot(1,4,4)
    plt.imshow(threshold_mask, cmap='gray');plt.title(f'Predicted Mask with cutoff={threshold}')
```

Figure B.22 Code to visualize the result of U-Net using matplotlib

Image Feature Extraction Software Code

The code in Figure B.23 creates the data loader for image feature extraction, Keras ImageDataGenerator lets us augment the images in real-time while the model is still training! it can apply any random transformations on each training image as it is passed to the model.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMAGE_SHAPE = (224, 224)
BATCH_SIZE = 32

train_dir = "/content/MMU_Iris_Dataset/train/"
test_dir = "/content/MMU_Iris_Dataset/test/"

train_datagen = ImageDataGenerator(rescale = 1/255.)
test_datagen = ImageDataGenerator(rescale = 1/255.)

print("Training images:")
train_data_10_percent = train_datagen.flow_from_directory(
    train_dir,
    target_size = IMAGE_SHAPE,
    batch_size = BATCH_SIZE,
    class_mode = "categorical")

print("Testing images:")
test_data = test_datagen.flow_from_directory(
    test_dir,
    target_size = IMAGE_SHAPE,
    batch_size = BATCH_SIZE,
    class_mode = "categorical")
```

Figure B.23 Creating data loaders in TL for iris image feature extraction

The code in Figure B.24 describes the model Created with the input image shape of

224,224, after getting a function for creating a model, I have used it to create a model using the ResNetV250 architecture as a feature extraction layer. After the model was instantiated, I compiled it using categorical_crossentropy as my loss function, the Adam optimizer, and accuracy as the metric.

```
[ ] def create_model(model_url, num_classes = 10):

    feature_extractor_layer = hub.KerasLayer(
        model_url,
        trainable = False,
        name = 'feature_extraction_layer',
        input_shape = IMAGE_SHAPE + (3,)) # define the input image shape

    model = tf.keras.Sequential([
        feature_extractor_layer, # use the feature extraction layer as the base
        layers.Dropout(0.4),
        layers.Dense(num_classes,
                      activation = 'sigmoid',
                      name = 'output_layer')
    ])

    return model
```

Figure B.24 Creating the Model in TL for iris image feature extraction

```
[ ] # Create model
resnet_model = create_model(resnet_url,
                            num_classes = train_data_10_percent.num_classes)

# Compile
resnet_model.compile(loss = 'categorical_crossentropy',
                    optimizer = tf.keras.optimizers.Adam(),
                    metrics = ['accuracy'])
```

Figure B.25 Creating the model with ResNet50V2 and compiling the model

I have got the training data ready in train_data_10_percent as well as the test data saved as test_data. Before I call the fit function, there's one more thing I have added, a callback. More specifically, a TensorBoard callback so it can track the performance of the model on TensorBoard. After 100 epochs, the ResNetV250 feature extraction model was able to obtain a good result with only 10 percent of the training images and test images. That goes to show the power of transfer learning. It's one of the main reasons whenever we are trying to model our own datasets, we should look into what pre-trained models already exist. The code is explained in Figure B.26.

```
[ ] from keras.applications import ResNet50V2
    from keras.api._v2.keras.layers import experimental
    # Fit the model
    resnet_history = resnet_model.fit(
        train_data_10_percent,
        epochs = 100,
        steps_per_epoch = len(train_data_10_percent),
        validation_data = test_data,
        validation_steps = len(test_data),

        callbacks = [create_tensorboard_callback(dir_name = "tensorflow_hub",
        experiment_name = "resnet50V2")])
```

Figure B.26 Fitting the ResNet50V2 model with 100 epochs

Image Classification Software Code

In iris image classification first I used the CNN model, and the model was performed with 3 convolution-pooling blocks, and the batch is 32, the input image size is (180,180), and the activation function have been used the 'relu' and 'softmax', the loss is 'sparse_categorical_crossentropy', the metrics are 'accuracy' and the optimizer is rmsprop, and with the epoch number of 100. The code is explained in Figure B.27.

```
[ ] inputs = keras.Input(shape=(180, 180, 3))
    x = layers.Rescaling(1./255)(inputs)
    x = layers.Conv2D(32, (3, 3), activation="relu")(x)
    x = layers.Dropout(0.2)(x)
    x = layers.MaxPooling2D(2, 2)(x)
    x = layers.Conv2D(64, (3, 3), activation="relu")(x)
    x = layers.Dropout(0.2)(x)
    x = layers.MaxPooling2D(2, 2)(x)
    x = layers.Conv2D(128, (3, 3), activation="relu")(x)
    x = layers.Dropout(0.3)(x)
    x = layers.MaxPooling2D(2, 2)(x)
    x = layers.Conv2D(256, (3, 3), activation="relu")(x)
    x = layers.Dropout(0.5)(x)
    x = layers.MaxPooling2D(2, 2)(x)
    x = layers.Flatten()(x)
    outputs = layers.Dense(25, activation="softmax")(x)
    cnn_model = keras.Model(inputs, outputs)

    cnn_model.compile(loss="sparse_categorical_crossentropy",
        metrics=["accuracy"],
        optimizer="rmsprop")

    hist = cnn_model.fit(train_ds, epochs=100, validation_data=val_ds)
```

Figure B.27 Creating, Compiling and Fitting the CNN image classification

The code in Figure B.28 Created train and test data generators and re-scale the data, then turned it into batches. The target_size defines the input size of dataset images in

(224, 224) format. The `class_mode` value of 'binary' defines the classification problem type. The `batch_size` defines how many images will be in each batch.

```
[ ] from tensorflow.keras.preprocessing.image import ImageDataGenerator
    train_datagen = ImageDataGenerator(rescale = 1/255.)
    test_datagen = ImageDataGenerator(rescale = 1/255.)

[ ] train_data = train_datagen.flow_from_directory(
    directory = train_dir,
    target_size = (224, 224),

    class_mode = 'binary',

    batch_size = 32)

test_data = test_datagen.flow_from_directory(
    directory = test_dir,
    target_size = (224, 224),
    class_mode = 'binary',
    batch_size = 32)
```

Figure B.28 Creating train and test data generators for classification

The 2D means the inputs are two-dimensional (height and width), even though they have 3 color channels, the convolutions are running on each channel individually. filters are the number of "feature extractors" that are moving over the images. After a model is created it needs to be compiled and fitted. `steps_per_epoch` is the number of batches a model will go through each epoch, in this case, I wanted the model to go through all batches so it's equal to the length of `train_data` and `validation_data`.

```
[ ] iris_model = Sequential([
    Conv2D(filters = 10,
          kernel_size = 3,
          strides = 1,
          padding = 'valid',
          activation = 'relu',
          input_shape = (224, 224, 3)),

    Conv2D(10, 3, activation = 'relu'),
    Conv2D(10, 3, activation = 'relu'),
    Flatten(),
    Dense(1, activation = 'sigmoid')
])
```

Figure B.29 Creating the Model with a 3-layer CNN on ImageNet

```
[ ] # Compile the model
iris_model.compile(loss = 'binary_crossentropy',
                  optimizer = Adam(),
                  metrics = ['accuracy'])

# Fit the model
hist = iris_model.fit(train_data,
                    epochs = 150,
                    steps_per_epoch = len(train_data),
                    validation_data = test_data,
                    validation_steps = len(test_data))
```

Figure B.30 Compiling and fitting the model created on ImageNet

The code in Figure B.31 Created the model with a 3 layer Convolutional Neural Network, the activation functions are 'relu' and 'sigmoid', in compiling the loss is defined as 'binary_crossentropy', optimizer as 'Adam' and metrics as 'accuracy'

```
[ ] TinyVGG_model = Sequential([
    Conv2D(10, 3, activation = 'relu', input_shape = (224, 224, 3)),
    MaxPool2D(pool_size = 2),
    Conv2D(10, 3, activation = 'relu'),
    MaxPool2D(),
    Conv2D(10, 3, activation = 'relu'),
    MaxPool2D(),
    Flatten(),
    Dense(1, activation = 'sigmoid')
])

[ ] TinyVGG_model.compile(loss = 'binary_crossentropy',
                        optimizer = Adam(),
                        metrics = ['accuracy'])

[ ] history_5 = TinyVGG_model.fit(train_data,
                                epochs = 150,
                                steps_per_epoch = len(train_data),
                                validation_data = test_data,
                                validation_steps = len(test_data))
```

Figure B.31 Creating, Compiling and fitting the model on TinyVGG

Appendix C – Biometric Method

Numerous biometrics researchers have trusted machine learning and deep learning algorithms for identification targets. Machine learning and deep learning algorithms need almost extraction techniques to extract features from raw biometric data and transform the raw data into a suitable and reliable format before classifying it. Furthermore, machine learning algorithms necessitate and involve certain preprocessing stages to be derived prior to feature extraction. Additionally, some extraction techniques do not continuously work positively with different types of biometrics or different data sets of the same biometrics. Moreover, they cannot handle biometric image transformations, for instance, zooming and rotation. In the interpretation of the countless performance of deep learning methods in several identification tasks, the structure of the proposed biometric identification system was based on convolutional neural networks (CNNs) which extract features and classify images by SoftMax, Sigmoid, and ReLU activation function classifier.

Biometrics are automated methods of identifying a person based on a physiological or behavioral characteristic [21]. Sometimes biometrics is the science of identifying or verifying the identity of a person based on physiological and behavioral characteristics. Biometric technologies are becoming the foundation of an extensive array of highly secure identification and personal verification solutions [158]. Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological versus behavioral characteristics [53]. A physiological biometric would identify by Hand, Face, Ear, Eye, fingerprint, and DNA. Behavioral biometrics is related to the behavior of a person. Some researchers have coined the term behavioral biometrics to describe the latter class of biometrics. [159] The selection of a particular biometric for use in a specific application involves a weighting of several factors that are as follows.

Table C.1 Seven Pillars Of Biometrics System [159]

1	Universality	Each and every person should have the characteristic such as fingers, iris, face, DNA which can be used for identification.
2	Distinctiveness	Every person should be sufficiently different in terms of the characteristics of another person.
3	Permanence	These characteristics should not largely change throughout human being life.
4	Collectability	Human characteristics can be measured and collected for quick identification.
5	Performance	The degree of accuracy and speed of identification must be quite high before the system can be operational.
6	Acceptability	Which indicates the extent to which people are willing to accept the use of a particular biometric characteristic in their daily lives.
7	Circumvention	In order to provide added security, a system needs to be harder to circumvent identity management systems.

Theoretical Background of Iris Biometric Methods

Iris identification or iris scanning is an automated method of biometric identification or the process of using visible and near-infrared light to take a high-contrast photograph of a person's iris which uses mathematical pattern recognition techniques and appears as one of the most useful methods for biometrics recognition and identification in the last few decades. Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristics [25]. The technology is mainly used for identification and access control or for identifying individuals who are under surveillance. The basic premise of biometric authentication is that every person can be accurately identified by intrinsic physical or behavioral traits [16]. The term biometrics is derived from the Greek words **bio**, meaning **life**, and **metric**, meaning to **measure**. The physiological characteristics are relatively stable, such as the face, fingerprint, hand silhouette, iris pattern, blood vessel pattern of the retina, or DNA fingerprint. Those biometric traits are essentially fixed and do not change over time. On the other hand, behavioral characteristics are more prone to change depending on factors such as aging, injuries, or even mood. There are so many techniques based on the seven factors provided in Table C.1.

Mathematical Methods

This section, explains mathematical operations and equations and analyzes, how to use methods in a mathematical process to calculate data and give accurate information in the form of mathematical numbers. Methods and algorithms are calculated, to develop an algorithm to automatically segment the iris region from an eye image, extract the features of images and then match the features, and the mathematical process is done to measure the speed training and testing the models.

In iris identification after image preprocessing the first phase of the iris image segmentation process, a method is employed to detect and remove reflections which is capable of influencing the iris recognition performance. Reflections when they occur within the iris portion are capable of introducing spurious information into the iris pattern, which can degrade the identification accuracy [158]. Also, when reflection occurs near the iris boundaries, they normally possess stronger properties than the properties of the pupil or iris edges which means that it is capable of influencing the segmentation algorithm and leading to erroneous segmentation. Therefore, it is very important that a fast and accurate algorithm for detecting and removing reflections without greatly altering the iris structure is needed.

Usually, specular reflection appears as the brightest points surrounded by large gradient values in iris image [160]. Given the iris image $I(x,y)$ where (x,y) denote the location of a pixel, the regions of highest intensities can be detected by the following inequality expression in Equation C.1 [161]:

$$I(x,y) > \beta_s \quad (C.1)$$

Where $I(x,y)$ denotes the intensity of the image and β_s is the brightness of the image which is given in Equation C.2:

$$\beta_s = \frac{\sum_{i=1}^N I(x,y)}{N} \quad (C.2)$$

Where N denotes the total number of pixels in the image calculated by multiplying

the row size by the column size. After that, a high-pass filtered iris image can be represented in Equation C.3:

$$f(x,y) = |\epsilon f(x,y)|^p Hf(x,y) \quad (C.3)$$

Where $|\epsilon f|^p$ is the image formed by raising every point $Lf(x,y)$ in the image ϵf to the p -th power. The image $|\epsilon f|^p$ weights the highpass filtered image Hf pointwise according to the strength of the local edges ϵ . Further details on highpass filtering can be found in [162]. The smoothed image $f(x,y)$ was decomposed by applying a discrete wavelet transform (DWT). At scale 2^j three detail images are obtained $W_{2^j}^h$, $W_{2^j}^v$ and $W_{2^j}^d$ and an approximation image W_{2^j} . The superscripts h , v , and d represent horizontal, vertical, and diagonal detail coefficients, respectively. Edges in the candidate reflection areas are characterized at each scale 2^j by computing the local maxima of the detailed coefficients which are given in Equation C.4 [163]:

$$D_{2^j}f(x,y) = \sqrt{|W_{2^j}^h f(x,y)|^2 + |W_{2^j}^v f(x,y)|^2 + |W_{2^j}^d f(x,y)|^2} \quad (C.4)$$

An edge-weighted highpass filtered image was computed by pointwise manipulation of highpass filtered image Hf with $|D_{2^j}f|$ which produced a $P + 1^{st}$ - order weighted highpass filters tuned to edges at scale 2^j . The multiscale analysis produced a set of edge-weighted highpass filters, each tuned to edges at prescribed scale. Therefore the reflection map $R(x,y)$ could be given as which represents the boundaries of the reflection points as computer in Equation C.5:

$$R(x,y) = \sum_{p=1}^P \sum_{j=1}^J |D_{2^j}f|^p Hf \quad (C.5)$$

Estimation of the Approximate Pupil Boundary: The pupil is the darkest portion of the eye image which is surrounded by the relatively lighter-colored iris. Consequently, there exists a large contrast between the dark pupil and the surrounding iris portion in most iris images. The pupil can sometimes be partially eclipsed by occlusions from the eyelids/eyelashes. A Hough transform is essentially a shape detector, which has been widely used to detect straight lines and circles in edge images [164]. The image

is first complemented before convolving it with a square matrix of size $n \times n$ of all ones to reduce noise. Circular Hough Transform was then applied to search for candidate pupils C_p in the image. The Euclidian distances of all the C_p from the center of the image are then computed. The C_p with the lowest Euclidean distance to the image center is then chosen as the pupil.

After, the adjusted intensity values I^{C_p} of the pupil candidates C_{pi} were calculated by convolving the intensity values of the image with a square matrix of size 5×5 of all ones in order to smoothen the image and reduce noise. Hough transform was employed to detect the pupil's candidates within the image. The actual pupil candidate was selected based on their distances from the image center and the intensity variation (entropy) within the candidate pupil circle. This could be stated mathematically as follows in Equation C.6:

$$I^{C_p}(x, y) = \frac{I'_{x,y}}{1+3d^2(x,y)} + H \quad (C.6)$$

Where d^2 represents the Euclidean distance measure between a pupil candidate and the center of the image. H is a measure of dispersion inside a candidate pupil C_p computed using Shannon entropy which is defined in Equation C.7:

$$H = \sum_{i=1}^m P_i \log \frac{1}{P_i} = \sum_{i=1}^m p_i \log P_i \quad (C.7)$$

I' is the adjusted intensity value which is given in Equation C.8:

$$I' = (255 - 1) * 1_{n \times n} \quad (C.8)$$

The segmentation process was achieved through the minimization of an energy function called graph energy. So, let the label $L = l_1, l_2, l_3, \dots, l_i, \dots, l_p$ where p is the number of pixels in the iris image and $l_i \in 0, 1$. The set L contained two labels 0, 1, where the pixels labeled 1 belong to the object while the pixels labeled 0 belong to the background. The energy function can be defined according to Equation C.9, which can be minimized using the max-flow/min-cut in the $s - t$ graph [165].

$$E(\psi) = \varphi E_B(\psi) + E_R(\psi), \quad (C.9)$$

E_B is the boundary energy term that incorporated the boundary constraint into the segmentation process while E_R is the regional energy term representing the regional image energy information in the segmentation process. φ is a regularization term between the regional and the boundary information which weights the importance of each in the equation, When φ was set to 0, the regional information was ignored and only the boundary information was considered. The regional energy term can be defined as in Equation C.10 [166]:

$$E_R = \sum_{p \in P} R_p(L_p) \quad (C.10)$$

Where $R_p(L_p)$ is the penalty for assigning the label l_p to the pixel p . The weight of $R_p(L_p)$ is obtained by comparing the intensity of the pixel p with the given intensity histogram of the object and background. The weights of the $t-link$ s is defined as follows in Equation C.11 and Equation C.12:

$$E_p(1) = -\ln Pr(I_p | 'Object'), \quad (C.11)$$

$$E_p(0) = -\ln Pr(I_p | 'background'), \quad (C.12)$$

Therefore, when the probability of the object $Pr(I_p | 'Object')$ is larger than that of the background $Pr(I_p | 'background')$. $E_p(1)$ is smaller than $E_p(0)$. Similarly, the boundary term $B_{<p,q>}$ was defined by the increasing function of $|l_p, l_q|$ as follows in Equation C.13 [167]:

$$B_{<p,q>} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2} \cdot \frac{1}{dist(p,q)}\right), \quad (C.13)$$

Where I_p and I_q represent neighboring pixels and σ notes the camera noise. This indicates that when the intensity difference between the two neighboring pixels is small, the penalty is high, otherwise it is low.

Now, consider an image that is embedded in a discrete grid. The neighborhood system N_g of the image can be of different forms, the edges of the neighborhood system form a family of lines in different directions $[-, /, |, \backslash]$. The cost of an s, t cut in the constructed graph is now equivalent to:

$$c(S, T) = \sum_i \sum_{j \in T} c(i, j) = \sum_{k=1}^{N_g} \alpha_n(k) \omega_k, \quad (\text{C.14})$$

Where $\alpha_c(k)$ is the number of arcs or edges of the family of lines k that connects S to T and ω_k is the weight of the arcs of the family of lines k .

Eyelashes are most times superimposed on the iris in non-ideal iris images. This can significantly hamper the recognition accuracy if not detected and excluded. The eyelash appears in the iris image as dark transient signals with high-intensity variation than the background. This can easily be measured using the traditional wavelet transformation. Let M, N be the original iris image which is then subdivided into blocks where each block is a sub-image of the original image. The high-frequency sub-bands of the detailed wavelet coefficients denoted by $D_i = \{HL, LH, HH\}$ are able to capture the high-frequency variations in both horizontal, vertical, and diagonal directions. The lowest computational cost wavelet transform, the Haar wavelet with one level of resolution is chosen to decompose each block into four frequency bands of size $\frac{n1}{2} \times \frac{m1}{2}$. The square root of the second order moment of wavelet coefficients in each detailed coefficient D_i is computed as defined in Equation C.15 [168].

$$f(x) = \left(\frac{1}{2} \sum_{i=0}^{\frac{Xl}{2}} \sum_{j=0}^{\frac{Yl}{2}} W_{k,i,j}^2 \right)^{\frac{1}{2}} \quad (\text{C.15})$$

Where $k \in D_i$ and W_k represent the matrix of wavelet coefficients of the block in the D_i set. The texture descriptor representing the maximum energy of the block (local energy) is determined in Equation C.16:

$$E_n = \max f(k), \forall k = D_i \quad (\text{C.16})$$

Where $f(k)$ is the energy of the high-frequency sub-bands. A 1-D energy vector E_n is then used in a block-wise clustering approach to capture the region of interest (ROI).

In order to compensate for the elastic deformation of the iris texture, the localized iris regions are unwrapped into a normalized rectangular block of a fixed size by converting from the Cartesian coordinate system to the polar coordinate system [169]. Given $I(x,y)$ is the localized image, the polar representation of the form $I(r,\theta)$ can be obtained as follows in Equation C.17 and Equation C.18:

$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \quad 0 \leq r \leq r_{max}, \quad (C.17)$$

$$\theta = \tan^{-1} \left(\frac{y - y_i}{x - x_i} \right), \quad (C.18)$$

Where r and θ are defined with respect to the center coordinates x_i, y_i . The center coordinate values obtained during the elliptical fitting are used as the center points for unwrapping the iris image.

After giving the normalized iris sample $N(i,j)$ it has been derived the principal components PC by maximizing the variances in the directions of the principal vectors. For the given normalized iris sample, with X, Y -dimensional vectors n_j aligned in the data matrix X . Let u be a direction which is a vector of length 1. The projection of the j -th vector n_j onto the vector u can be computed using Equation C.19 given as:

$$P_j = \vec{u}^T \cdot \vec{n}_j \sum_{i=1}^M u_i n_{i,j} \quad (C.19)$$

It has been found a direction u that maximized the variance of the projections of all the input vectors $t_j, j = 1, \dots, N, i = 1, \dots, M$. The maximization function was given as follows in Equation C.20:

$$J^{PCA} \vec{u}^T = \sigma^2(P_j) = \frac{1}{N} \sum_{j=1}^N (P_j - \bar{P})^2 = \dots \vec{u}^T \sum \vec{u}, \quad (C.20)$$

Where $\sum$ is the covariance matrix of the data matrix N . Let $N_{i,j}$ denote a normalized

iris image. The PCA can be given as in Equation C.21:

$$\Sigma = \frac{1}{N} \hat{X} \hat{X}^T \quad (\text{C.21})$$

From the covariance matrix, Σ the eigenvalues and eigenvectors were calculated. The eigenvalues and eigenvectors represent the direction of the principal texture pattern. The largest eigenvalue equals the maximal variance, while the corresponding eigenvector determines the direction of the maximal variance. For instance, the 1,1 entry of Σ is given as in Equation C.22:

$$\Sigma_{11} = \frac{1}{3-1} ((a_1, \mu_1)^2 + (b_1, \mu_1)^2 + (c_1, \mu_1)^2) \quad (\text{C.22})$$

Which is precisely the variance of the first variable. Also, the 2, 1, entry of the covariance Σ can be given as in Equation C.23:

$$\Sigma_{11} = \frac{1}{3-1} ((a_1, \mu_1)(a_2, \mu_2) + (b_1, \mu_1)(b_2, \mu_2) + (c_1, \mu_1)(c_2, \mu_2)) \quad (\text{C.23})$$

The eigenvector corresponding to the highest eigenvalue is the principal eigenvector and its angle can be given as Equation C.24:

$$\theta = \arctan \frac{\lambda_i - \Sigma_{11}}{\Sigma_{21}} \quad (\text{C.24})$$

Where λ_i represents the eigenvalues. Once the principal texture orientation has been computed, a complex wavelet filter is constructed in order to elicit important iris information in the direction of the principal texture.

The discrete wavelet transform (DWT) involves the evaluation of the scale and shift parameters on a discrete grid of time-scale plane which leads to a discrete set of continuous basis functions. The time-scale plane can be indexed using two integers, where the first integer, j corresponds to the discrete scale step while the second integer, n corresponds to the discrete translation step. The discretization is performed by setting the dilation parameter $s = s_0^j$ and the translation parameter $u = k s_0^j u_0$, for $j, k \in \mathbb{Z}$. Where, $s_0 > 1$, represents the dilated step, and $u \neq 0$ represents the translated step. The

family of wavelets can then be given as Equation C.25:

$$(\psi)_{j,n}(t) = s_0^{-j/2} \psi(s_0^{-j} t - k u_0 s_0^j) \quad (C.25)$$

Therefore, the discrete wavelet decomposition of a function $f(t)$ can be given as Equation C.26:

$$f(t) = \sum_j \sum_k D_f(j,n) \psi_{j,n}(t), \quad (C.26)$$

Where $D_f(j,n)$ represents the 2-D set of coefficients of the function and $\psi_{j,k}(t)$ is the wavelet basis function (mother wavelet).

The discretization of the wavelet basis function is such that $s_0 = 2$ and $u_0 = 1$ s in the dyadic time-scale grid. Then, the selection of $\psi(t)$ representing the discrete wavelet transform is done in such a way that basis function set $\{\psi_{j,k}\}$ constitute an orthonormal basis of $L^2(\mathbb{R})$ such that Equation C.27:

$$D_f(j,n) = \int_{-\infty}^{\infty} \psi_{j,n}(t) f(t) dt = \langle \psi_{j,n}(t) f(t) \rangle \quad (C.27)$$

This represents the wavelet basis for the evaluation of the function $f(t)$ using the summation of finite basis over index j and n with finite DWT coefficients.

The iris sample contains a huge number of features and a comparatively small number of samples per class, and this makes accurate and fast classification or recognition difficult [170]. Given an iris feature vector pool S the position of a particle s is given as $S_i = (s_{i,1}, s_{i,2}, s_{i,3}, \dots, s_{i,N})^T$, and the velocity $V_i = (v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,N})^T$. The particle's flight experience is given as $P_i = (p_{i,1}, p_{i,2}, p_{i,3}, \dots, p_{i,N})^T$ which is regarded as the *pbest* and the group experience is $G_i = (g_{i,1}, g_{i,2}, g_{i,3}, \dots, g_{i,N})^T$ which is also regarded as the *gbest*. The particle is able to determine the next movement through its own experience *pbest* and the group experience of the entire population *gbest*. For the $k+1^{th}$ iteration, each particle of PSO changes position according to the following Equation C.28:

$$V_{i,j}^{k+1}(t, 1) = \omega V_{i,j}^k + c_1 r_1 * (P_{lbest} - s_{i,j}^k) + c_2 r_2 * (P_{gbest} - s_{i,j}^k), \quad (C.28)$$

Where $i = 1, 2, 3, \dots, M$ and M is the number of particles; $V_{i,j}^k$ is the d th dimensional component of the k th iteration, the velocity vector of the particle s and $s_{i,j}^k$ is the d th iteration dimensional component of the k th iteration of the position vector of particle s . p_{best} is the d th dimensional component of the best position of an individual particle s . Then the classification function is given by Equation C.29:

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i . y_i . x_i + b\right) \quad (\text{C.29})$$

Now, for the iris image, the data cannot be linearly separated. Therefore, the kernel function has been used to transform that data into a higher dimensional space. The classification foundation was then transformed to Equation C.30:

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i . y_i . k(x_i + x) + b\right) \quad (\text{C.30})$$

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Ahmad SALIHY

Date of Birth :

Phone :

E-mail :

EDUCATION

High School : Abdul Rahim Shahid High School, Kabul, Afghanistan.

Bachelor : Kateb University, Kabul, Afghanistan. Apr 2012 – Nov 2015

Master Degree : ANKARA YILDIRIM BEYAZIT UNIVERSITY

WORK EXPERIENCE

Software Developer : Shahid Vahdat Maili Water Supply Company

Shahid Vahdat Maili Water Supply Company, District 13th, Kabul, Afghanistan

Jan 2015 - Aug 2022

TOPICS OF INTEREST

- | | |
|---|---|
| - Artificial intelligence | - Machine learning |
| - Computer vision | - Human-computer interaction |
| - Algorithms and data structures | - Security and Privacy |
| - Massive Big Data Processing of Software and Tools | - Bioinformatics and scientific computing |

PRESENT ORGANIZED SOURCES / PRESENTATION