Menoufia University
Faculty of Electronic Engineering
Department of Computer Science and Engineering

Iris Recognition for Internet of Things Security

A Thesis Submitted for the Degree of M. Sc. in Engineering Science
Computer Science and Engineering
Artificial Intelligence and Image Processing
Department of Computer Science and Engineering

by

Eng. Ahmed Sabry Abd Elkhalek Shalaby

Teaching Assistant, Computer Science and Engineering Department
Faculty of Electronic Engineering, Menoufia University, Egypt.

B. Sc. in Electronic Engineering, Computer Science and Engineering
Faculty of Electronic Engineering, Menoufia University, Egypt. (2017)

Supervised by

Prof. Dr. Nawal Ahmed El- Fishawy

Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University

Dr. Ezz Eldin Badawy Gad Alrab Hemdan

Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University

2021

Menoufia University
Faculty of Electronic Engineering
Department of Computer Science and Engineering

Iris Recognition for Internet of Things Security

A Thesis Submitted for the Degree of M. Sc. in Engineering Science
Computer Science and Engineering
Artificial Intelligence and Image Processing
Department of Computer Science and Engineering

by

Eng. Ahmed Sabry Abd Elkhalek Shalaby

Teaching Assistant, Computer Science and Engineering Department
Faculty of Electronic Engineering, Menoufia University, Egypt.

B. Sc. in Electronic Engineering, Computer Science and Engineering
Faculty of Electronic Engineering, Menoufia University, Egypt. (2017)

Supervised by

Prof. Dr. Nawal Ahmed El- Fishawy **( )**
Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University

Dr. Ezz Eldin Badawy Gad Alrab Hemdan **( )**
Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University

2021

Menoufia University
Faculty of Electronic Engineering
Department of Computer Science and Engineering

Iris Recognition for Internet of Things Security

A Thesis Submitted for the Degree of M. Sc. in Engineering Science
Computer Science and Engineering
Artificial Intelligence and Image Processing
Department of Computer Science and Engineering

by

Eng. Ahmed Sabry Abd Elkhalek Shalaby

Teaching Assistant, Computer Science and Engineering Department
Faculty of Electronic Engineering, Menoufia University, Egypt.

B. Sc. in Electronic Engineering, Computer Science and Engineering
Faculty of Electronic Engineering, Menoufia University, Egypt. (2017)

Approved by

Prof. Dr. Nawal Ahmed El- Fishawy **( )**
Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University

Prof. Dr. Amany Mahmoud Sarhan **( )**
Professor and Head of Computer and Control Engineering
Department, Faculty of Engineering, Tanta University

Assoc. Prof. Nirmeen A. El-Bahnasawy **( )**
Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University

Vice Dean for the Postgraduate Studies and Research

Prof. Dr. Mona Shokeir

2021

Acknowledgment

To my supervisors- namely Prof. Dr. Nawal El-Fishawy, Dr. Ezz Eldin Badawy. A special thank you to Dr. Ramadan Gad for his support and efforts, no simple appreciation or thank you can describe all the efforts you made to finish this thesis.

I would like to give a very special thanks to my family - for helping me all the time.

List of Publications

A list of published work during the creation of this thesis:

1. Ahmed Shalaby, *et.al*., " An efficient multi-factor authentication scheme based CNNs for securing ATMs over cognitive-IoT." PeerJ Computer Science Journal. https://doi.org/10.7717/peerj-cs.381
2. Ahmed Shalaby, *et.al*., " "An Efficient CNN based Encrypted Iris Recognition Approach in Cognitive-IoT System", Multimedia Tools and Applications Journal.

Abstract

Recently, biometric-based security plays a vital role in the success of the Internet of Things (IoT) based security framework. The iris trait solves a lot of security issues, especially in smart IoT-based applications. It increases the resistance of these systems against severe authentication attacks. In this thesis, an efficient iris recognition system based on deep Convolutional Neural Networks (CNNs) is proposed for IoT applications. CNN is used to extract the deep iris features from the left and right eyes, which will be used as input features to a fully connected neural network with a Softmax classifier. CASIA V4 Interval dataset, Phoenix dataset, and UBIRIS V1 are used to train the CNN system; to get the best tuning of network parameters. The results showed that the proposed approach attains supreme accuracy compared to the existing approaches, it is obtained up to 98%, 99.24%, and 100% with UBIRIS V1, CASIA V4, and Phoenix datasets, respectively. The proposed system achieves satisfied and competitive results regard accuracy, and robustness among existing methods. Likewise, the proposed method has a relatively low training time, which is a useful parameter in critical IoT-based applications.

In this thesis, we propose a practical extension to the later iris recognition system in a bigger system for a generic IoT environment built using the raspberry pi 2 kit and its accessories. Also, the effect of adding different kinds of noise to iris images, due to noise interference related to sensing IoT devices, bad acquisition of iris images by system users, or other system assaults, is discussed. The effect of two different types of noise is discussed, the first being randomly generated from Gaussian distribution and the second from the Uniform distribution. Chaotic encryption is utilized to secure the transmission of iris templates in the proposed system. Regards to recognition accuracy rate for this extension of the proposed iris recognition system, this methodology shows low degradation of recognition accuracy rates in the case of using noised iris images.

One of the most important environments of IoT is Automatic Teller Machines (ATMs). Because the identity verification of banks' clients at Automatic Teller Machines (ATMs) is a very critical task. Clients' money, data, and crucial information need to be highly protected. The classical ATM verification method using a combination of credit card and password has a lot of drawbacks like Burglary, robbery, expiration, and even sudden loss. However, integrating an efficient iris recognition system in critical IoT environments like ATMs may involve many complex scenarios. To address these issues, this thesis proposes a practical novel efficient full authentication system for ATMs based on a bank’s mobile application and a visible light environments-based iris recognition.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AES	Advanced Encryption Standard
ALT	Adaptive Local Threshold
AMQP	Advanced Message Queuing Protocol
ATMs	Automatic Teller Machines
CDF	Cumulative Density Function
CNNs	Convolutional Neural Networks
CoAP	Constrained Application Protocol
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DWT	Discrete Wavelet Transform
E-banking	Electronic banking
FCNN	Fully Connected Neural Network
HD	Hamming Distance
HTTP	Hypertext Transfer Protocol
ID	Identity / Identification
IoT	Internet of Things
IWT	Integer Wavelet Transform
LFSR	Linear Feedback Shift Register
MQTT	Message Queue Telemetry Transport
NIR	Near-Infrared
OTP	One Time Password
PCA	Principal Component Analysis
PDF	Probability Density Function
PNN	Probabilistic Neural Network
RSA	Rivest–Shamir–Adleman
SVM	Support Vector Machine
VGG-Net	Visual Geometry Group at the University of Oxford
XMPP	Extensible Messaging and Presence Protocol

List of Symbols

$\mathbf{Rmatrix}$	Summation matrix for each row
$\mathbf{Cmatrix}$	Summation matrix for each column
$\mathbf{R_C}$	Row-centroid
$\mathbf{C_C}$	Column-centroid
$\mathbf{P_C(x, y)}$	Coordinate of pupil center
$\mathbf{R_p}$	Pupil radius
$\mathbf{M_p(x, y)}$	Pupil mask
$\mathbf{M_e(x, y)}$	Eyelashes mask
$\mathbf{M_o(x, y)}$	Occlusion mask
$\mathbf{A_{RR}}$	Accuracy of recognition rate
$\mathbf{N_c}$	Number of correct classifications
$\mathbf{T_c}$	Total Number of classifications
$\mathbf{R_{test}^{train}}$	Ratio between training to testing datasets
$\gg$	Shift left operator
$\oplus$	XOR operator
$\boldsymbol{\mu}$	Mean value of a random variable.
$\boldsymbol{\sigma}$	Standard deviation a random variable

Chapter 1

Introduction

In the last years, most of our daily applications are based on IoT systems where several devices and sensors are connected. These applications produce a huge amount of data. The IoT applications are considered the source of the biggest kind of information on the internet, so identity verification becomes a very crucial and very challenging task when it has to be automated with high accuracy of recognition rates and low probability of break-ins in IoT systems [1]. Biometrics plays a great role in solving these security problems nowadays. Iris as a biometric has a lot of advantages, as we will see later, to make it one of the best biometrics to help us achieve our goals and efficiently face these security problems. However, integrating an efficient iris recognition system in critical IoT environments like Automatic Teller Machines (ATMs) may involve many complex scenarios. This work presented in the thesis addresses some of these issues. In this chapter, we give a broad introduction to the work presented in this thesis as follows: Section 1.1 gives an overview of authentication systems, biometrics, and iris biometric. Section 1.2 discusses the motivations of this thesis and the main problems that motivated it. Section 1.3 focuses on the main contributions of the thesis. Finally, section 1.4 contains the organization of the thesis.

1.1 Overview

Person authentication [1] is of course an old problem that has a broad history, and the community has adopted several methods to verify the identity of persons. There are three main adopted ways of authentication [2], as shown in figure 1.1:

1) Possessions: mainly depends on possessions of physical objects like door keys, credit cards, and passports.
2) Knowledge: mainly depends on knowing important information which is considered to be kept as a secret and is only known to the legitimate person. For example, passwords, secret words, and passphrases.
3) Biometrics [3]: mainly depends on the physiological and behavioral characteristics of persons that distinguish one person from others. Physiological biometrics such as iris, face, or hand geometry, are physical characteristics that are measured at some point in time. Behavioral biometrics such as gait, lip motion, or voice, on the other hand, consists of the way each person carrying out his or her action and extends over time.

Figure 1.1: The three basic ways a person can prove identity.

The first two adopted ways of authentication have some drawbacks like identification cards may be lost, forged, or misplaced and passwords may be forgotten or compromised. The conventional methods of identification, based on possessions of physical objects like door keys or credit cards, or based on exclusive knowledge of important information which is considered to be kept as a secret such as passwords or even passphrases are not sufficient, reliable, or secure of critical IoT applications. So, it is clear that any reliable positive person identification must entail biometric identification. The main advantage of biometrics over the first two methods is that it cannot be misplaced, forgotten, or stolen. Also, it is very difficult to spoof biometric traits [4]. Due to the greater accuracy and higher robustness of biometric recognition [5]; Biometric solutions have become popular and preferred methods to analyze human characteristics for security - authentication and identification – purposes [6], [7].

Among other biometrics, iris [8] is considered to be the most accurate, stable, and reliable biometric technology available today. It has a lot of desirable characteristics like:

- Uniqueness.
- Stability with age.
- Not genetically connected, unlike eye color.
- Impossible to alter surgically.
- Externally visible.
- highly protected.
- Works on blind persons.

The three methods of authentication, discussed above, can be used in isolation or combination [9]. A combination of authentication methods generates what is so-called multi-factor authentication. In biometrics we distinguish two authentication methods:

1. Verification is based on a combination of a unique identifier which singles out a particular person like an ID number and that a biometric of that person.
2. Identification is based only on biometric measurements.

1.2 Thesis Motivation and Problem Statement

Iris recognition problem is a special kind of pattern recognition [10],[11] or classification problem that has a very long history [12], in which we need to make a classifier to a set of classes of data. There is a need to extract the best features that represent an individual’s iris, to facilitate the role of the classifier, and reduce its complexity. Designing handcrafted feature extractors for biometric data is a very complex and challenging task [13]. It takes a lot of time and it needs a great knowledge of the field that governs these data, and it is not guaranteed to achieve high accuracy of recognition rates.

Deep learning [14][15] came into the picture, especially Convolutional Neural Networks (CNNs) which can give us a very good understanding of image data without depending completely on any domain knowledge and handcrafted features.

A lot of researchers [16][17], who addressed the use of CNNs with iris traits, used pre-trained models of CNNs like VGG-16 [18], ResNet50 [19], and Inceptionv3 [20]. These models are trained on a very large number of data classes that exclude iris classes themselves and using these models as a black box. So, using these pre-trained models is also not guaranteed to achieve high accuracy recognition rates because of the loss of the biometric information which was not used in the training stage of building these models. It is not suitable in a lot of cases to modify these huge pre-trained models to satisfy our needs because of the restrictions of training time and available space.

Building a new iris-based CNN model, as the proposed model in this thesis, needs an intelligent selection of the number of kernels, the kernels’ dimensions, input image dimensions, and other factors that affect the model recognition rate [21]. It also needs a huge number of experiments of training and testing to achieve the best architecture that has a high recognition accuracy rate with relatively low training time [22].

Many existing systems for iris recognition achieve a well-accepted recognition rate [23], [24]. But the majority of them deals with iris acquired by infrared or near-infrared cameras [25], which are completely not suitable for a large class of IoT applications that mainly depend on usual light vision cameras.

Even with using iris biometric-based IoT environments like ATMs, the communication channels are still a weak point in the overall system. Any penetration of these channels endangers the system. So, iris encryption is crucial here, but conventional cryptography techniques like AES, RSA, and DES [26], [27]are not suitable for biometric data due to inseparable characteristics of biometric data like high correlation among adjacent pixels, high redundancy, etc. [28]. The chaotic theory is favorable for the encryption of biometric data; as it is very sensitive to initial conditions, pseudorandom in nature, and has high resistivity against system attacks [29].

In general, any iris recognition system will not be embedded in an ideal environment concerning iris acquisition. Iris images may gain some external noise, at the iris acquisition stage, due to several reasons like system attacks, environmental dust, interference noise on iris sensing devices, or non-appropriate iris acquisition due to faults of system users, different illumination states …. etc.

1.3 Thesis Contribution

The contributions of this thesis can be summarized as follows:

- Proposing an efficient system via utilizing the right and left iris images from both eyes for providing a strong iris-based authentication system for IoT applications with confirming the person through the right and left irises. This iris recognition system depends on handcrafted deep CNN as a feature extractor, and a fully connected neural network (FCNN) - with Softmax layer - as a classifier. It can be more reliable and favored than a lot of state-of-the-art methods that are used in building iris recognition systems.
- Evaluating the proposed system via various experimental datasets that were captured using different iris acquisition conditions which helps the proposed system to be more suitable for a large class of IoT applications. Two public datasets Phoenix [30], [31],[32] and UBIRIS. V1 [33], [34] captured using usual light vision cameras. It is also evaluated via datasets acquired by near-infrared cameras which is the CASIA V4 dataset [25].
- Providing a secure method to address the problem of hacking the iris template transmission over the communication channels of IoT systems by protecting the iris using an encryption algorithm based on a chaotic key sequence generated by the sequence of the logistic map and sequence of states of Linear Feedback Shift Register (LFSR)[35].

- Studying and discussing the effect of several kinds of noise from different distributions [36],[37] on iris images. It is due to noise interference or bad acquisition or any other system attacks.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2 provides a literature review about biometrics, IoT-based iris recognition, and fundamentals of deep learning.
- Chapter 3 focuses on the basic components of the proposed iris recognition system in general without diving into applying it in IoT environments, describes the methodologies used for iris segmentation, iris normalization, feature extraction, and classification, and finally discusses the related results of these methodologies.
- Chapter 4 discusses the extension of the proposed system in a generic IoT environment, describes the used communication methodology, discusses iris encryption and decryption over communication channels, and studies the effect of adding different kinds of noise to iris images to evaluate the reliability of the proposed system.
- Chapter 5 introduces a real-life scenario from IoT environments which is ATMs for banking systems and discusses how to embed the proposed iris recognition system in it.
- Chapter 6 summarizes the conclusions for the proposed system and intended future works.

Chapter 2

Theoretical Background and Literature Review

In this chapter, we give a literature overview about the work presented in this thesis as follows: Section 2.1 gives an overview of biometrics, and iris biometric. Section 2.2 gives an overview of the Internet of Things (IoT) and its applications. Section 2.3 focuses on deep learning especially Convolutional Neural Networks (CNNs). Finally, section 2.4 contains the related works of the thesis.

2.1 Biometrics and Iris Recognition

Biometric systems are constantly evolving and promise technologies that can be used in automatic systems for identifying or authenticating a person’s identity uniquely and efficiently without the need for the user to carry or remember anything, unlike traditional methods like passwords, IDs [38]. More precisely, biometrics is the science of identifying or verifying the identity of, a person based on physiological or behavioral characteristics.

Physiological biometrics, like fingerprints or hand geometry, are physical characteristics generally measured at some point in time. Behavioral biometrics, like signature or voice, on the other hand, consists of the way some action is carried out and extends over time. Behavioral biometrics are learned or acquired over time and are dependent on one's state of mind or even subject to deliberate alteration. The distinction between physiological and behavioral biometrics in the later definition of biometrics is useful. But we should also keep in mind other attributes that are necessary to make a biometric practical. These include the five properties [39]:

1) Universality: Every person should have the biometric characteristic.
2) Uniqueness: No two persons should be the same in terms of the biometric characteristic.
3) Permanence: The biometric characteristic should be invariant over time.
4) Collectability: The biometric characteristic should be measurable with some (practical) sensing device.

A property perhaps less inherently connected with a particular biometric is:

5) Acceptability: The particular user population and the public, in general, should have no (strong) objections to the measuring collection of the biometric.

In this regard, iris recognition has been utilized in many critical applications, such as access control in restricted areas, database access, national ID cards, and financial services, and is considered one of the most reliable and accurate biometric systems [40]. Several studies have demonstrated that the iris trait has several advantages over other biometric traits (e.g., face, fingerprint), which make it commonly accepted for application in high reliability and accurate biometric systems. Firstly, the iris trait represents the annular region of the eye lying between the black pupil and the white sclera; this makes it completely protected from varied environmental conditions [41]. Secondly, it is believed that the iris texture provides a very high degree of uniqueness and randomness, so it very unlikely for any two iris patterns to be the same, even irises from identical twins, or the right and left eyes of a person. This complexity in iris patterns is due to the distinctiveness and richness of the texture details within the iris region, including rings, ridges, crypts, furrows, freckles, zigzag patterns [42]. Thirdly, the iris trait provides a high degree of stability during a person's lifetime from one year of age until death. Finally, it is considered the most secure biometric trait against fraudulent methods and spoofing attacks by an imposter where any attempt to change its patterns, even with surgery, is a high risk, unlike the fingerprint trait which is relatively easier to tamper with. Despite these advantages, implementing an iris recognition system is considered a challenging problem due to the iris acquisition process possibly acquiring irrelevant parts, such as eyelids, eyelashes, pupil, and specular reflections which may greatly influence the iris segmentation and recognition outcomes.

2.2 Internet of Things

Today the Internet has become ubiquitous, has touched almost every corner of the globe, and is affecting human life in unimaginable ways. However, the journey is far from over. We are now entering an era of even more pervasive connectivity where a very wide variety of appliances will be connected to the web. We are entering an era of the Internet of Things (IoT). This term has been defined by different authors in many different ways. Let us look at two of the most popular definitions. Vermesan et al. [43] define the Internet of Things as simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors and actuators. Another definition in [44] defines the Internet of Things as a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object. We use these capabilities to query the state of the object and to change its state if possible. In common language, the Internet of Things refers to a new kind of world where almost all the devices and appliances that we use are connected to a network. We can use them collaboratively to achieve complex tasks that require a high degree of intelligence. For this intelligence and interconnection, IoT devices are equipped with embedded sensors, actuators, processors, and transceivers. IoT is not a single technology; rather it is a clustering of various technologies that work together in tandem.

Sensors and actuators are devices, which help in interacting with the physical environment. The data collected by the sensors have to be stored and processed intelligently to derive useful inferences from it. Note that we broadly define the term sensor; a mobile phone or even a microwave oven can count as a sensor as long as it provides inputs about its current state. An actuator is a device that is used to effect a change in the environment such as the temperature controller of an air conditioner.

The storage and processing of data can be done on the edge of the network itself or a remote server. If any preprocessing of data is possible, then it is typically done at either the sensor or some other proximate device. The processed data is then typically sent to a remote server. The storage and processing capabilities of an IoT object are also restricted by the resources available, which are often very constrained due to limitations of size, energy, power, and computational capability. IoT devices are characterized by low resources in computation and energy capacity [45]. As a result, the main research challenge is to ensure that we get the right kind of data at the desired level of accuracy.

Along with the challenges of data collection, and handling, there are challenges in communication as well. The communication between IoT devices is mainly wireless because they are generally installed at geographically dispersed locations. The wireless channels often have high rates of distortion and are unreliable. In this scenario reliably communicating data without too many retransmissions is an important problem and thus communication technologies are integral to the study of IoT devices.

There are many requirements for IoT. It can be summarized the requirements into these key categories [46]:

1) Device management: each IoT device has configurable parameters depending on embedding system requirements, these parameters should be managed remotely.
2) Communications and connectivity; IoT system nodes should be connected to do the system tasks.
3) Data processing: data should be collected and stored for processing and then taking actions based on the analysis.
4) Security: is an important aspect that is intended for the IoT. IoT devices are often collecting personal data, and by the normal operation, they are bringing the real world onto the internet which means that security is a must.

5) Highly available: the system should be available at all the running cases and does not fail.
6) Scalability: an important requirement for the IoT server-side architecture, a system can support a huge number of devices.
7) Prediction: each task required by the IoT system should be handled with predefined specific time requirements.

There is no single consensus on architecture for IoT [47], which is agreed upon universally. Different architectures have been proposed by different researchers. The most basic architecture is a three-layer architecture [48][49] as shown in Figure 2.1. It was introduced in the early stages of research in this area.

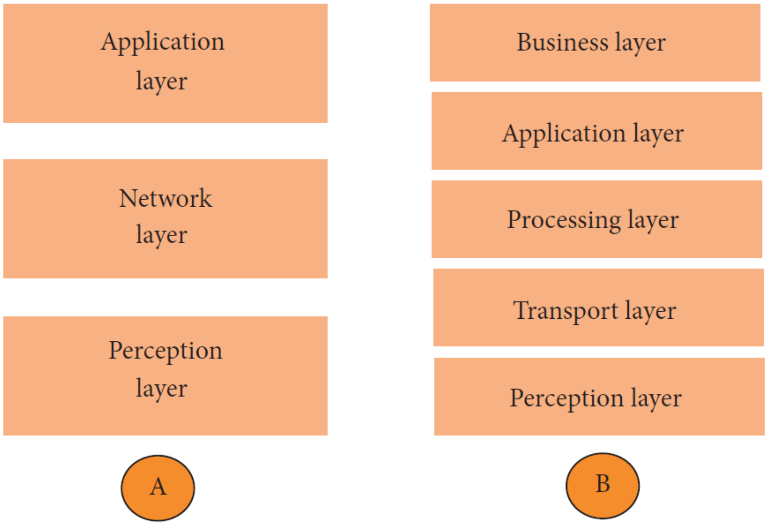


Figure 2.1: Architecture of IoT (A: three layers) (B: five layers).

It has three layers, namely, the perception, network, and application layers.

1) The perception layer is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.
2) The network layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

3) The application layer is responsible for delivering application-specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.

The three-layer architecture defines the main idea of the Internet of Things, but it is not sufficient for research on IoT because research often focuses on finer aspects of the Internet of Things. That is why we have many more layered architectures proposed in the literature. One is the five-layer architecture, which additionally includes the processing and business layers [50][51]. The five layers are perception, transport, processing, application, and business layers as shown in Figure 2.1. The role of the perception and application layers is the same as the architecture with three layers. The function of the remaining three layers is as follows:

1) The transport layer transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID, and NFC.
2) The processing layer is also known as the middleware layer. It stores, analyzes, and processes huge amounts of data that comes from the transport layer. It can manage and provide a diverse set of services to the lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules.
3) The business layer manages the whole IoT system, including applications, business and profit models, and users' privacy.

IoT has a wide range of applications, the IoT top applications [52] such as smart cities, smart environment, smart water, smart metering, security and emergencies, retail, logistics, industrial control, smart agriculture, smart animal farming, home automation, and E-Health.

There are many application domains based on the internet of things, the applications could be classified based on the network type, coverage, availability, scale, repeatability, heterogeneity, user involvement, and impact. Application domains are classified into four domains [53]:

1) Personal and home: The collected information of sensors is handled only by the individuals who own the network directly. Wi-Fi is usually used as a backbone for enabling higher bandwidth data for video transfer and higher sampling rates for sound. As an extension of the personal area, the network is deploying a home monitoring system, and the home equipment control such as air refrigerators, conditioners, washing machines, etc.
2) Enterprise: A network of things within a work environment is called an enterprise-based application in which information collected is used only by the owners. The first common application is environmental monitoring which is implemented to track and manage the utilities within the building such as lighting. This area includes the effect on citizens such as health and well-being issues; transport considering its impact on productivity, mobility, pollution; and services in terms of community services which is managed and provided by the local government to inhabitants of the city.
3) Utilities: The network information in this application domain is for service optimization, not consumer consumption. It is used by utility companies for example the smart meter by electricity supply companies for resource management to optimize the cost vs. profit. Usually, these areas are used by a large organization on a regional and a national scale. These are very extensive networks to monitor the critical utilities and to manage the resource efficiently. The backbone network used could be Wi-Fi, cellular, and satellite communication. Smart metering and smart grid, video-based applications, water monitoring, and water drinking quality assurance are good examples for the utility application domain.
4) Mobile: Smart logistics and smart transportation are considered in the mobile domain due to the data sharing nature and the required backbone implementation. Dynamic traffic information would have an effect on the freight movement allowing improved scheduling and better planning, the transport application enables the use of a large scale of wireless sensor networks for travel times online monitoring, and traffic control.

2.3 Convolutional Neural Network

Convolutional neural networks are designed to work with grid-structured inputs, which have strong spatial dependencies in local regions of the grid. The most obvious example of grid-structured data is a 2-dimensional image. This type of data also exhibits spatial dependencies because adjacent spatial locations in an image often have similar color values of the individual pixels. An additional dimension captures the different colors, which creates a 3-dimensional input volume. Therefore, the features in a convolutional neural network have dependencies among one another based on spatial distances. Other forms of sequential data like text, time-series, and sequences can also be considered special cases of grid-structured data with various types of relationships among adjacent items. The vast majority of applications of convolutional neural networks focus on image data, although one can also use these networks for all types of temporal, spatial, and spatiotemporal data. An important property of image data is that it exhibits a certain level of translation invariance, which is not the case in many other types of grid-structured data. For example, a banana has the same interpretation, whether it is at the top or the bottom of an image. Convolutional neural networks tend to create similar feature values from local regions with similar patterns. One advantage of image data is that the effects of specific inputs on the feature representations can often be described intuitively.

An important defining characteristic of convolutional neural networks is an operation, which is referred to as convolution. A convolution operation is a dot-product operation between a grid-structured set of weights and similar grid-structured inputs drawn from different spatial localities in the input volume. This type of operation is useful for data with a high level of spatial or other locality, such as image data. Therefore, convolutional neural networks are defined as networks that use the convolutional operation in at least one layer, although most convolutional neural networks use this operation in multiple layers.

A Convolutional Neural Network is a feed-forward multilayer neural network, which differs from traditional fully connected neural networks by combining several locally connected layers aimed at automated feature recognition, followed by several fully connected layers aimed at classification [54]. The CNN architecture, as illustrated in Figure 2.2, comprises several distinct layers including sets of locally connected convolutional layers (with a specific number of different learnable kernels in each layer), subsampling layers named pooling layers, and one or more fully connected layers. The internal structure of the CNN combines three architectural concepts, which make the CNN successful in different fields, such as image processing and pattern recognition, speech recognition, and NLP.

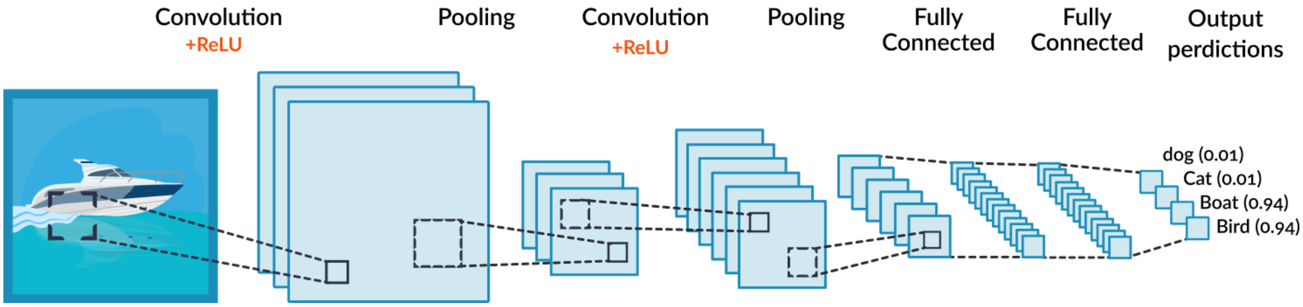


Figure 2.2: An illustration of the CNN architecture [22].

The first concept is applied in both convolutional and pooling layers, in which each neuron receives input from a small region of the previous layer called the local receptive field [55] equal in size to a convolution kernel. This local connectivity scheme ensures that the trained CNN produces strong responses to capture local dependencies and extracts elementary features in the input image (e.g., edges, ridges, curves, etc.) which can play a significant role in maximizing the inter-class variations and minimizing the intra-class variations and hence increasing the Accuracy of Recognition Rate (ARR) of the iris recognition system. Secondly, the convolutional layer applies the sharing parameters (weights) scheme to control the model capacity and reduce its complexity. At this point, a form of translational invariance is obtained using the same convolution kernel to detect a specific feature at different locations in the iris image [56]. Finally, the nonlinear downsampling applied in the pooling layers reduces the spatial size of the convolutional layer’s output and reduces the number of the free parameters of the model. Together, these characteristics make the CNN very robust and efficient at handling image deformations and other geometric transformations, such as translation, rotation, and scaling [14]. In more detail, these layers are:

- Convolutional layer: In this layer, the parameters are organized into sets of 3-dimensional structural units, known as filters or kernels. The filter is usually square in terms of its spatial dimensions, which are typically much smaller than those of the layer the filter is applied to. On the other hand, the depth of a filter is always same is the same as that of the layer to which it is applied. Assume that the dimensions of the filter in the qth layer are $(F_q * F_q * d_q)$. An example of a filter with $F_1 = 5$ and $d_1 = 3$ is shown in Figure 2.3.

  The convolution operation places the filter at each possible position in the image (or hidden layer) so that the filter fully overlaps with the image and performs a dot product between the $(F_q * F_q * d_q)$ parameters in the filter and the matching grid in the input volume (with same size $(F_q * F_q * d_q)$). The dot product is performed by treating the entries in the relevant 3-dimensional region of the input volume and the filter as vectors of size $(F_q * F_q * d_q)$, so that the elements in both vectors are ordered based on their corresponding positions in the grid-structured volume. An example of a convolution operation is shown in Figure 2.4.

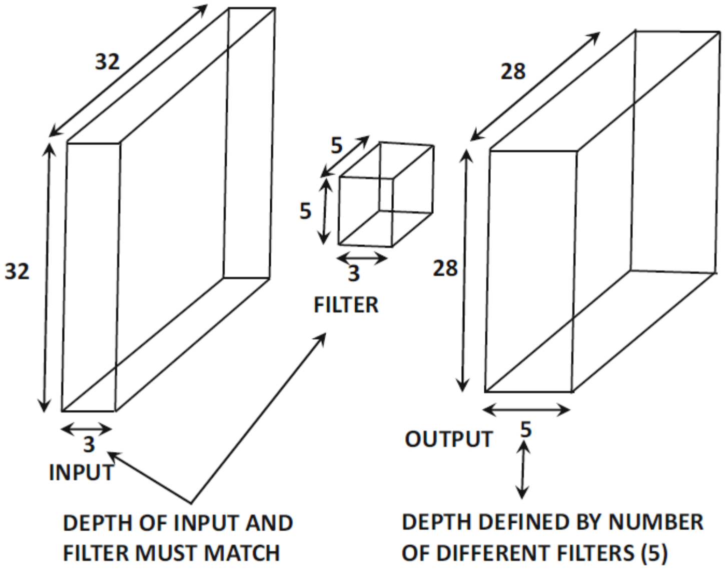


Figure 2.3: The convolution between an input layer of size 32 × 32 × 3 and a filter of size 5 × 5 × 3 produces an output layer with spatial dimensions 28 × 28 [22].

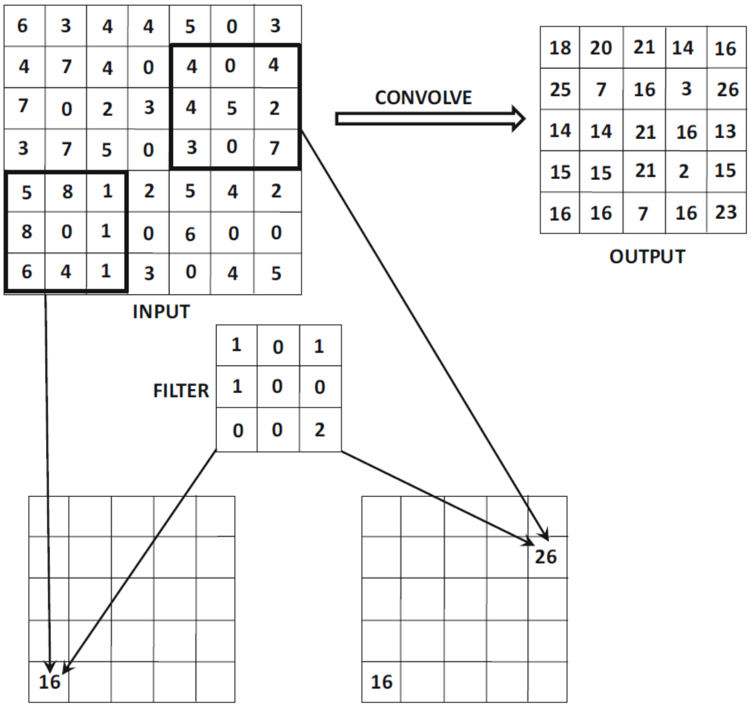


Figure 2.4: An example of a convolution between a 7 × 7 × 1 input and a 3 × 3 × 1 filter with stride of 1. A single filter will always create a single feature map irrespective of its depth [22].

- Pooling layer: Its main function is to reduce the spatial size of the convolutional layers' output representations, and it produces a limited form of translational invariance. Once a specific feature has been detected by the convolutional layer, only its approximate location relative to other features is kept. The pooling operation is, however, quite different. The pooling operation works on small grid regions of size $P_q * P_q$ in each layer and produces another layer with the same depth. For each square region of size $P_q * P_q$ in each of the d_q activation maps, the maximum of these values is returned. This approach is referred to as max-pooling. If a stride of 1 is used, then this will produce a new layer of size $(L_q - P_q + 1) * (B_q - P_q + 1) * d_q$. However, it is more common to use a stride $S_q > 1$ in pooling. In such cases, the length of the new layer will be $\frac{(L_q - P_q)}{S_q} + 1$ and the breadth will be $\frac{(B_q - P_q)}{S_q} + 1$ Therefore, pooling drastically reduces the spatial dimensions of each activation map. Unlike with convolution operations, pooling is done at the level of each activation map. Whereas a convolution operation simultaneously uses all $d_q$feature maps in combination with a filter to produce a single feature value, pooling independently operates on each feature map to produce another feature map. Therefore, the operation of pooling does not change the number of feature maps. In other words, the depth of the layer created using pooling is the same as that of the layer on which the pooling operation was performed. Examples of pooling with strides of 1 and 2 are shown in Figure 2.5.

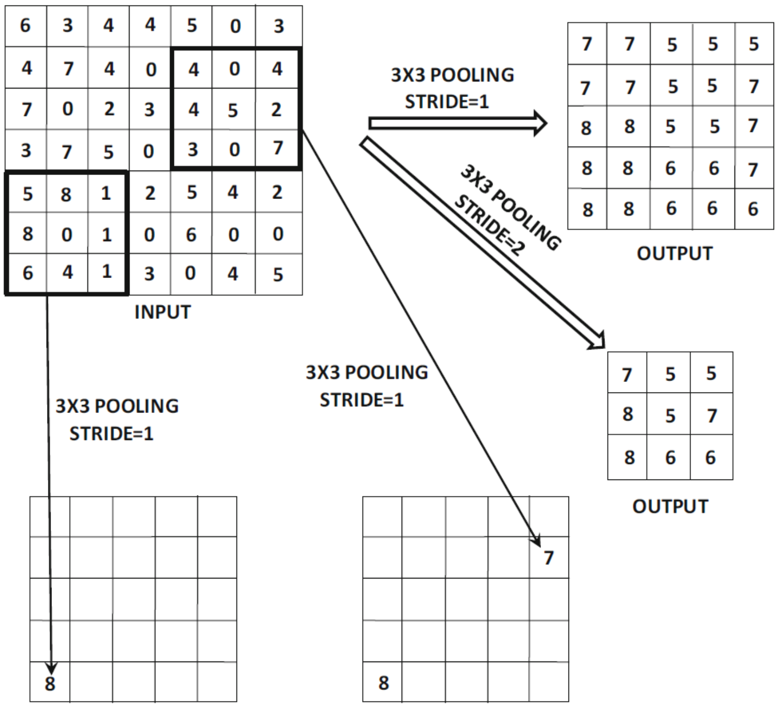


Figure 2.5: An example of a max pooling of one activation map of size 7 × 7 with strides of 1 and 2. A stride of 1 creates a 5 × 5 activation map. A stride of 2 creates a 3×3 activation map [22].

- Fully connected layers: Each feature in the final spatial layer is connected to each hidden state in the first fully connected layer. This layer functions in the same way as a traditional feed-forward network. In most cases, one might use more than one fully connected layer to increase the power of the computations towards the end. The connections among these layers are exactly structured like a traditional feed-forward network. Since the fully connected layers are densely connected, the vast majority of parameters lie in the fully connected layers. For example, if each of the two fully connected layers has 4096 hidden units, then the connections between them have more than 16 million weights. Similarly, the connections from the last spatial layer to the first fully connected layer will have a large number of parameters.

2.4 Literature Review

In 1993, the first successful and commercially available iris recognition system was proposed by Daugman [57]. In this system, the inner and outer boundaries of the iris region are detected using an integro-differential operator. Afterward, the iris template is transferred into a normalized form using Daugman's rubber sheet method. This is followed by using a 2D Gabor filter to extract the iris features and the Hamming distance for decision making. However, as reported in [58][59][60], the key limitation of Daugman's system is that it requires a high-resolution camera to capture the iris image and its accuracy significantly decreases under non-ideal imaging conditions due to the sensitivity of the iris localization stage to noise and different lighting conditions. In addition to Daugman, many researchers have proposed iris recognition systems using various methods, among which the most notable systems were proposed by Wildes [61], Boles and Boashash [62], Lim et al. [63], and Masek [64]. However, most existing iris recognition systems claim to perform well under ideal conditions using developed imagery set up to capture high-quality images, but the recognition rate may substantially decrease when using non-ideal data. Therefore, the iris recognition system is still an open problem and the performance of the state-of-the-art methods still has much room for improvement.

In past years, several works have been done in the domain of iris recognition. Mainly the researchers differ in the way of extracting features from iris images. A lot of them used handcrafted feature extractors to build their classification systems.

In the literature, several publications have documented the high accuracy and reliability of traditional neural networks. However, traditional neural networks have several drawbacks and obstacles that need to be overcome. Firstly, the input image is required to undergo several different image processing stages, such as image enhancement, image segmentation, and feature extraction to reduce the size of the input data and achieve satisfactory performance. Secondly, designing a handcrafted feature extractor needs good domain knowledge and a significant amount of time. Thirdly, an MLP has difficulty in handling deformations of the input image, such as translations, scaling, and rotation [65]. Finally, a large number of free parameters need to be tuned to achieve satisfactory results while avoiding the overfitting problem. A large number of these free parameters is due to the use of full connections between the neurons in a specific layer and all activations in the previous layer [66]. To overcome these limitations and drawbacks, the use of deep learning techniques was proposed, and some researchers addressed the use of CNN as a feature extractor.

A lot of researchers [17][16][67], who addressed the use of CNNs with iris traits, used a pre-trained model of CNNs like VGG-16 [18], ResNet50 [19], Inceptionv3 [20], and AlexNet [68]. These pre-trained models are trained on a very large number of data classes, that exclude iris classes themselves and using these models as a black box. So, using such pre-trained models they are not warranted to achieve high accuracy recognition rates; because of the biometric information loss which was not used in the training stage of these models.

An iris recognition system is proposed in [67], where the authors used the pre-trained model of Xception as a feature extractor. Then, they used the Pre-trained model DeepLabV3+ with MobileNet for classification. They tested their model against CASIA Thousand dataset. They achieved a 97.46% accuracy of recognition rate, which is considered a relatively good recognition rate, but it could be better without using these generic pre-trained models.

An iris recognition system is proposed in [16], where they used the pre-trained model of Visual Geometry Group at the University of Oxford (VGG-Net) as a feature extractor. Then, they used a multi-class Support Vector Machine (SVM) algorithm for classification. They tested their model against the CASIA-Iris-Thousand dataset. They achieved a 90% accuracy of recognition rate, which is considered a relatively moderate recognition rate due to using a pre-trained model and the loss of biometric information during training.

The authors in [17], proposed an iris recognition system, where they used the pre-trained Alex-Net model as a feature extractor. Then, they used a multi-class SVM algorithm also for classification. They tested their model CASIA-Iris-Interval dataset. They achieved 89% accuracy of the recognition rate, which is considered also a relatively moderate recognition rate due to using a pre-trained model and the loss of biometric information during training. An iris recognition system is proposed in [69], they applied a method used Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT), as a handcrafted method for extracting features and Euclidean Distance for classification. They tested their model against the Phoenix dataset. They have an average of 88.5% of recognition rate. And their pre-processing stage of their model does not include iris segmentation and normalization which affects their recognition rate.

The authors in [70], used Discrete Wavelet Transform (DWT) with Principal Component Analysis (PCA), as a handcrafted method for extracting features and Support Vector Machine (SVM) as a classifier. They tested their model against CASIA-Iris-V4. They get 95.40% as the accuracy of the recognition rate. A proposed iris recognition system in [71], with Haralick texture for extracting features and probabilistic neural networks (PNN) as a classifier. They tested their model against UBIRIS. V1. dataset, and get 97% accuracy of recognition rate, which is considered a Fairly good recognition rate.

The iris recognition system proposed in [72], in which researchers used Radon transform and gradient-based isolation as a handcrafted method for extracting features and Euclidean distance for classification. They tested their model against the CASIA-iris-V3 dataset. The accuracy of the recognition rate is 84.17%, which is considered a relatively low recognition rate that will make the system not suitable for critical C-IoT applications.

The authors in [73], proposed an iris recognition system, where they used Haar wavelet and Daubechies wavelet for feature extraction. Then, they used a feedforward neural network as a classifier. They tested their model against the CASIA-Iris-V1 dataset. They achieved a 94.76% accuracy of the recognition rate, which is considered a relatively moderate recognition rate.

The researchers in [74], proposed an iris recognition system, where they used Integer Wavelet Transform (IWT) for feature extraction. Then, they used normalized Hamming distance as a classifier. They tested their model against the UBIRIS.v2 [33] dataset. They achieved a 98.9% accuracy of the recognition rate, which is considered a good recognition rate but the UBIRIS.v2 dataset is based only on one eye, which makes it less suitable for critical C-IoT applications.

The authors in [75] proposed an iris recognition system where they used the intensity of iris images as a feature extraction method and Hamming Distance (HD), Feed Forward Neural Network, and SVM for classification. They tested their model against the CASIA-iris-V3 dataset. They have 76.8%, 87%, and 98.5%, respectively, as the accuracy of the recognition rate for each classification method in a relatively low number of dataset classes which was 40 classes.

Some researchers [76], who applied iris biometric in IoT applications, do not apply any encryption technique for iris images before transmission, and this problem may put the system at risk of attacks. So, there is a lack of research work that has addressed the problem of building strong and efficient full authentication systems for IoT-based applications based on both left and right irises, which includes safe protection methods against attacks on communication networks.

Also, most of the researchers [17][74][69][75] who worked in iris recognition build their classification models based only on one human iris either for left or right iris. This limitation in real-life systems will decrease the system's reliability against attacks, which we resolve in the proposed model. The performance of the proposed system is evaluated with an accuracy metric for the recognition rate over the used two data sets and outperformed the previous work. Table 2.1 summarizes the results of most of the state-of-the-art methods discussed before.

TABLE 2.1
SUMMARY OF THE DISCUSSED RELATED WORKS

Approach	Dataset	Feature extraction	Classification	Recognition accuracy %
Ghanapriya Singha et al. (2020) [74]	UBIRIS.V2	Integer Wavelet Transform (IWT)	Normalized Hamming distance	98.9
Ruqaiya Khanam et al. (2019) [73]	CASIA-Iris-V1	Haar wavelet and Daubechies wavelet	feedforward neural network	94.76
Peter Peer et al. (2019) [67]	CASIA Thousand	Pre-trained Xception	Pre-trained DeepLabV3+ with MobileNet	97.46
HK Rana et al. (2019) [70]	CASIA-Iris-V4	PCA and DWT	SVM	95.40
Maram and Lamiaa (2018) [17]	CASIA-Iris-Interval	pre-trained Alex-Net model	Multi-Class SVM	89
Alaa Al-Waisy et al. (2018) [24]	CASIA-Iris-V3	Convolutional Neural Network	Softmax classifier + fusion	100
Minaee et al. (2016) [16]	CASIA-Iris-Thousand	pre-trained VGG-Net	Multi-Class SVM	90
Saminathan et al. (2015) [75]	CASIA-Iris-V3-interval	Intensity image	Least square method of quadratic SVM	98.50
S. S. Dhage et al. (2015) [69]	Phoenix	Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT)	Euclidean distance	88.50
Bharath et al. (2014) [72]	CASIA-Iris-V3	Radon transform and gradient-based isolation	Euclidean distance	84.17
Sundaram et al. (2011) [71]	UBIRIS. V1	Haralick features	Probabilistic Neural Networks (PNNs)	97

2.5 Chapter Summary

In this chapter, we gave a literature overview of all the theoretical backgrounds needed for this thesis. We discussed biometrics and their main categories (physiological and behavioral). We focused on the iris trait and its main characteristics which make commonly accepted for application in high reliability and accurate systems. Also, we discussed the internet of things, its main requirements, and its architecture. Then, we discussed convolutional neural networks and their main layers (convolutional, Pooling, and fully connected layers). Finally, we showed the state-of-art in the iris recognition field by discussing many recently related works.

Chapter 3

Proposed Iris Recognition System

In this chapter, we are proposing an efficient iris recognition system via utilizing both the right and left iris images for providing a strong iris-based authentication system that will be adapted later for IoT applications. This iris recognition system depends on handcrafted deep CNN as a feature extractor, and a fully connected neural network (FCNN) - with Softmax layer - as a classifier. The proposed system can be more reliable and favored than a lot of state-of-the-art methods that are used in building iris recognition systems. The Evaluation of the proposed system is done via various experimental datasets that were captured using different iris acquisition conditions (near-infrared cameras or usual light vision cameras) which help the proposed system to be more suitable for a large class of IoT applications.

In this chapter, we discuss the proposed iris recognition system as follows: Section 3.1 introduces the overall system structure. Section 3.2 gives a detailed description of the used iris datasets during experiments. Section 3.3 focuses on iris segmentation and normalization methodologies. Section 3.4 focuses on iris feature extraction and classification based on CNNs. Section 3.5 presents the results and discussions.

3.1 Iris Recognition system

The proposed iris recognition system, as shown in Figure 3.1, consists of three main stages as follows:

1) The pre-processing stage consists of iris segmentation and normalization.
2) Deep feature extraction using CNNs.
3) Classification using a fully connected neural network with a Softmax layer.

Before discussing these steps in detail, we give a full description of the used iris datasets during experiments because the nature of the dataset is involved in discussing the internals of these steps.

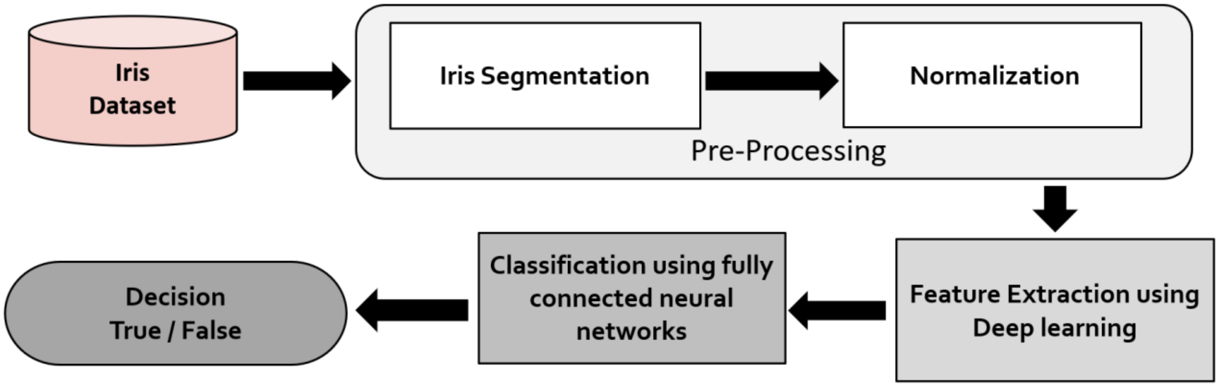

Iris Dataset
Iris Segmentation
Normalization
Pre-Processing
Feature Extraction using Deep learning
Classification using fully connected neural networks
Decision True / False

Figure 3.1: Overall structure of the proposed iris recognition system.

3.2 Iris Datasets

During all experiments in this work, to evaluate the proposed approach, three publicly available datasets are used, called CASIA V4-interval [25], Phoenix [30], [31], [32], and UBIRIS V1 [33], [34] respectively as the following:

1) CASIA V4-interval dataset is the latest dataset captured by CASIA self-developed close-up iris camera, all iris images are 8-bit Gray-level (. JPEG) files, collected under near-infrared (NIR) illumination. It consists of 2641 iris images taken from 249 subjects. Its iris images with resolution (320*280) pixel. This dataset has two problems. The first problem is that there are a lot of subjects without any iris image for either left or right iris, the second problem is that there are a lot of subjects that have a very small number of iris images for either left or right iris. These problems limited our choice of subjects used in the proposed system.

2) Phoenix dataset consists of 384 irises Image taken from 64 subjects, 192 for the left iris and 192 for the right iris, the iris images are 24-bit RGB of (. PNG) file format. Its iris is imaged with a resolution of 576 x 768 pixels. The irises were taken by TOPCON TRC50IA optical device connected to SONY DXC-950P 3CCD camera.

3) UBIRIS V1 dataset is composed of 1877 images collected from 241 persons during September 2004 in two distinct sessions. Its main characteristic result from the fact that, in opposition to the existing public and free databases (CASIA and Phoenix), it incorporates images with several noise factors, thus permitting the evaluation of robustness iris recognition methods. Three versions of this dataset are available depending on the resolution. the selected one is the gray-scale version of (. JPEG) file format. The resolution of its images is 200 x 150 pixels. The iris images were taken by Nikon E5700 camera with a Focal Length of 71 mm and Exposure Time of 1/30 sec.

The iris images in these datasets are captured under different situations of pupil dilation, eyelids/eyelashes occlusion, the slight shadow of eyelids, specular reflection, etc. Different samples from these datasets are shown in Figure 3.2 and the summary of the configuration of these datasets is shown in Table 3.1.

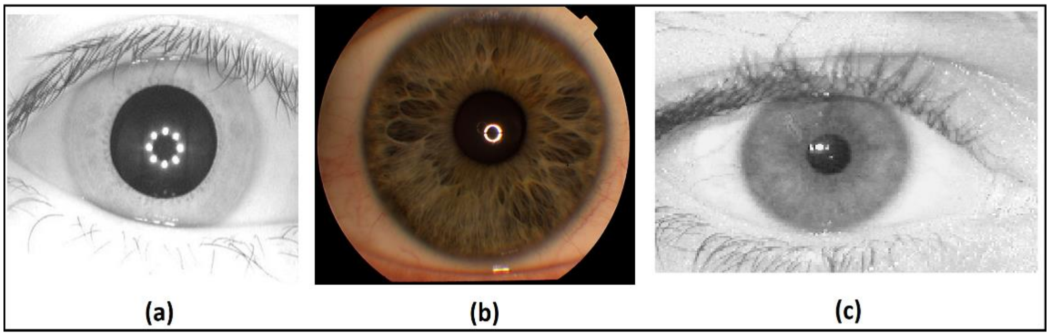


Figure 3.2: Samples from used datasets. (a) CASIA V4-interval dataset, (b) Phoenix dataset, (c) UBIRIS V1 dataset

TABLE 3.1
DESCRIPTION OF THE USED DATASETS.

	Visible light vision datasets		Near-Infrared (NIR) datasets
	Phoenix	***UBIRIS. V1***	***CASIA V4 interval***
Description	384 iris Images taken from 64 subjects	1877 images collected from 241 subjects	2641 iris image taken from 249 subjects
Image's resolution	576*768 pixels	200*150 pixels	320*280 pixels
Image format	(.PNG)	(.JPEG) (Check)	JPEG
Subjects used in localization	ALL	ALL	ALL
Subjects used in classification	60	50	55
Samples per subject	3 right and 3 left	5	not regular

3.3 Iris segmentation and normalization

In this subsection, we will illustrate the suggested method of generating the iris templates from the original iris image in this stage $A\ (x, y)$.

3.3.1 Pupil Detection for CASIA V4 and UBIRIS V1 datasets

As shown in [77], for removing the corneal and specular reflections in iris images of CASIA V4-Interval and UBIRIS V1 datasets, a sequence of morphological operations was proposed. For pupil detection, the Adaptive Local Threshold (ALT) algorithm which depends on the mean filter is used to filter bright pixels in the original iris image $A(x, y)$. Regards the result binary image, shown in Figure 3.3, let $Rmatrix$, $Cmatrix$ be the summation matrices for each row and each column of that binary image, respectively. The row-centroid (R_C) and column-centroid (C_c) are computed as shown in Equations 3.1 and 3.2. Where the '$Index$' parameter is the coordinate of the pixel in the image along the x-axis and y-axis. The sign '$|.|$' refers to the absolute value. The pupil center point ($P_C(x, y)$) is the intersection point of (R_C) and (C_c) The pupil radius (R_p) calculated [77] as follows:

$$R_C = \ Index\ (max|Rmatrix|) \tag{3.1}$$

$$C_C = \ Index\ (max|Cmatrix|) \tag{3.2}$$

$$R_p = max\ (R_{p1}, R_{p2}, R_{p3}, R_{p4}) \tag{3.3}$$

$$R_{p1} = \ \left|Index\ \left(P_1(x,y)\right) - \ Index\ \left(P_C(x,y)\right)\right| \tag{3.4}$$

$$R_{p2} = \left|Index\ \left(P_2(x,y)\right) - \ Index\ \left(P_C(x,y)\right)\right| \tag{3.5}$$

$$R_{p3} = \left|Index\ \left(P_3(x,y)\right) - \ Index\ \left(P_C(x,y)\right)\right| \tag{3.6}$$

$$R_{p4} = \ \left|Index\ \left(P_4(x,y)\right) - \ Index\ \left(P_C(x,y)\right)\right| \tag{3.7}$$

Each of the points P_1, P_2, P_3 and P_4 has the index of the first zero-valued pixel along the radius axis in the four directions. Pupil detection steps with the results are illustrated in Figure 3.4, in sequence. The pupil region mask $(M_p(x,y))$, as shown in Figure 3.4-d, identified by the center $\left(P_C(x,y)\right)$and the radius (R_p) is multiplied again in the original image $A(x,y)$; to isolate the iris region $I(x,y)$ free of artifacts without deformation. The pupil border is shown in Figure 3.4-f.

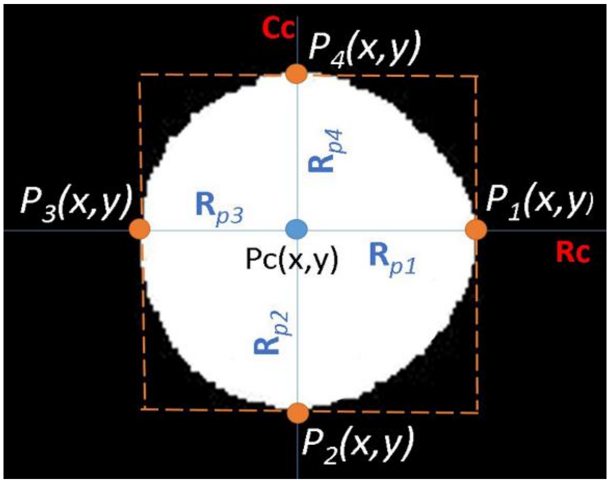


Figure 3.3: Pupil parameter detection (pupil center $\boldsymbol{P_C(x,y)}$ and pupil radius R_p).

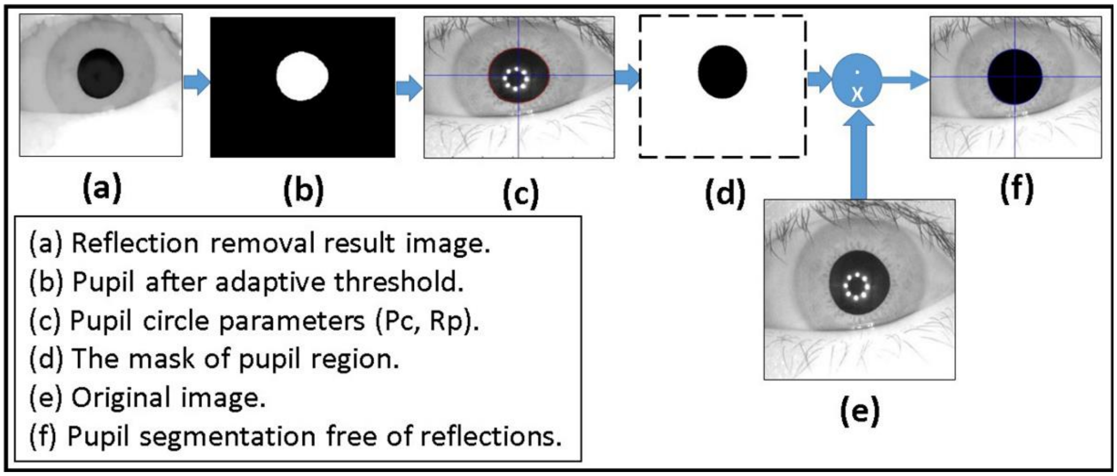

(a)
(b)
(c)
(d)
(e)
(f)
x
(a) Reflection removal result image.
(b) Pupil after adaptive threshold.
(c) Pupil circle parameters (Pc, Rp).
(d) The mask of pupil region.
(e) Original image.
(f) Pupil segmentation free of reflections.

Figure 3.4: Pupil detection steps.

3.3.2 Masking Technique

With the aid of the pupil detection parameters (P_1, P_2, P_3 and P_4) and the pupil region mask ($M_p(x,y)$), the iris mask could be declared and detect. First, the mask to isolate the eyelashes and eyelid's part calculated ($M_e(x,y)$). Then, multiply this mask as a binary matrix to the pupil mask generated ($M_p(x,y)$) . This multiplication will generate the final iris mask ($M_o(x,y)$) pixels that could multiply in the original image to localize the iris region free of pupil, eyelashes, and eyelids.

Let iris image $I\ (x,y)$ has $m \times n$ pixels, $\forall y\ 1 \leq y \leq n$, the eyelashes/eyelids removing a mask ($M_e(x,y)$) identified as shown in Equation 3.8:

$$M_e(x,y) = \begin{cases} 0: & 1 \leq x \leq Index\big(P_4(x,y)\big), Index\big(P_2(x,y)\big) \leq x \leq m \\ 1: & Index\big(P_4(x,y)\big) < x < Index\big(P_2(x,y)\big), \end{cases} \quad (3.8)$$

Here the mask has two binary values. Binary (0) declare each point of the eyelashes and eyelids up and down the pupil circle. And binary (1) represents the pupil and iris pixels between the eyelashes and eyelid parts.

The final mask ($M_o(x,y)$), is shown in Figure 3.5 and Equation 3.9, identified in a binary format as follows:

$$M_o(x,y) = M_p(x,y) * M_e(x,y) \quad (3.9)$$

The concatenation of the two masks represents the output iris mask, that used to isolate the iris region by the multiplication process of ($M_o(x,y)$) and the original image $A\ (x,y)$.

In [78], a fixed template size (60x90 pixel) generated. This was unsuitable for some images in datasets, due to image sizes and resolution modifications were done over MT; the N pixels to the left and right of the localized pupil are concatenated. The iris template is created by mapping the selected pixels on a fixed size (60×2 N) matrix as shown in Figure 3.6 (b-c).

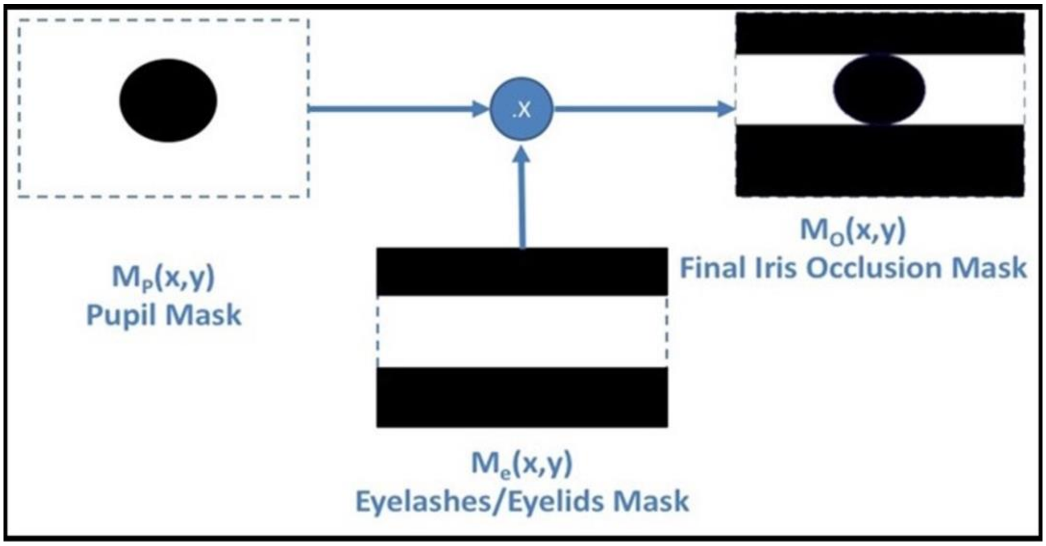


Figure 3.5: MT mask generation process

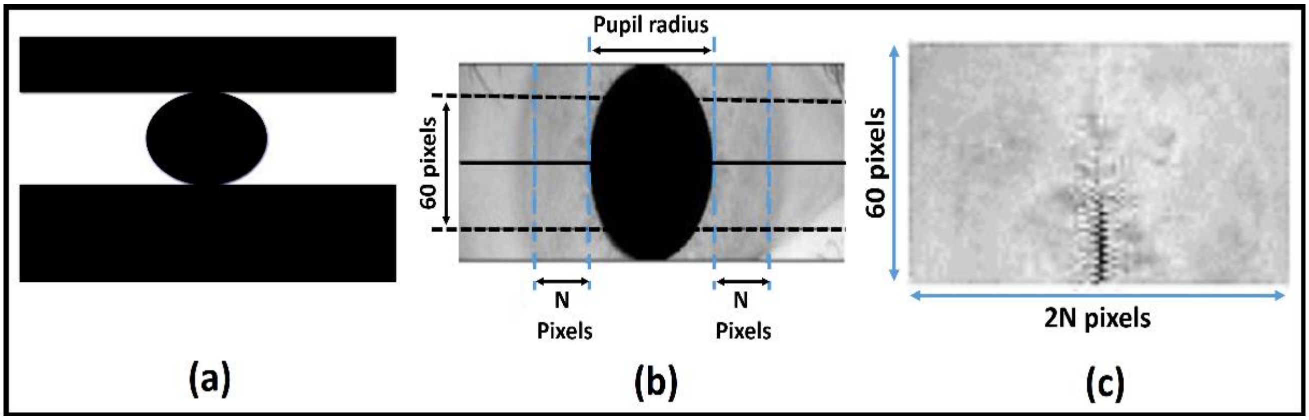


Figure 3.6: MT processes illustration. (a) Iris mask $\boldsymbol{M_o(x,y)}$ by the aid of pupil parameters and eyelashes mask. (b) Iris template parameter declaration. (c) Final iris template.

3.3.3 Iris template generation for the Phoenix dataset

In the case of the Phoenix dataset, the dataset is available already segmented as shown in Figure 3.7-a. In our experiments, First the upper half and the lower half of iris images were separated, each of the dimensions of (350*100) pixels. Then, they were concatenated together, as shown in Figure 3.7(b-d). The final image of dimensions of (350*200) is used in the next stage of feature extraction. This solution does not consider the rotation of both the camera and the eye. Moreover, it is suitable for offline images only.

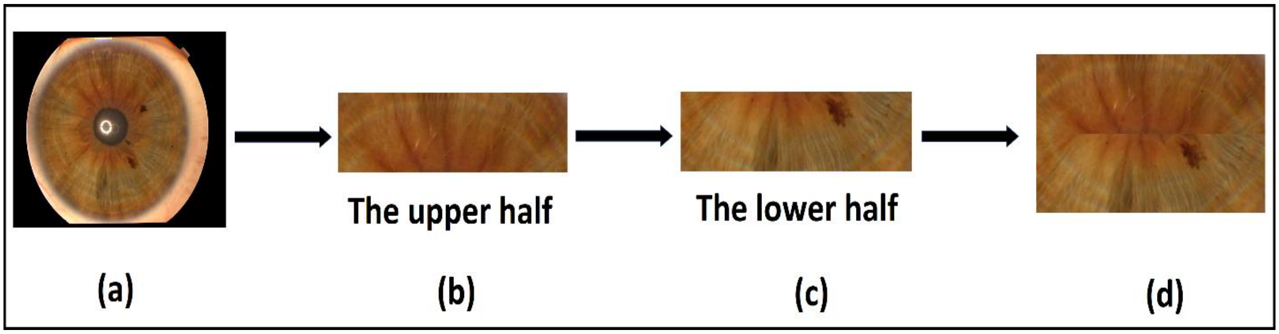


Figure 3.7: The proposed template generation for the Phoenix dataset.

3.4 CNN-based Deep Feature Extraction and Classification

Different CNNs [79] with different architectures, as it will be shown in the results section, were used to extract the deep features from iris images for the dual iris in the case of CASIA V4 and Phoenix datasets and the lonely iris in the case of the UBIRIS.V1 dataset. The goal was to find a more accurate architecture that increases the recognition accuracy rate with a relatively low training time. After these experiments, we propose a CNN model for all datasets. It consists of “3” convolutional layers, “3” max-pooling layers, “3” RELU activation layers, “2” fully connected layers, and “1” SoftMax layer as the architecture of the proposed model. Each CNN architecture tested; to maintain its recognition accuracy rate; to adjust the model configuration to achieve a higher recognition rate. The proposed architecture is illustrated in Table 3.2 and Figure 3.8.

TABLE 3.2
THE PROPOSED CNN ARCHITECTURE FOR ALL DATASETS

Layer name	No of filters	Filter size	Stride size	Padding
Conv1	100	3*3	1*1	Valid
RELU	n/a	n/a	n/a	n/a
Max pooling	1	2*2	2*2	Valid
Conv2	150	3*3	1*1	Valid
RELU	n/a	n/a	n/a	n/a
Max pooling	1	2*2	2*2	Valid
Conv3	200	3*3	1*1	Valid
RELU	n/a	n/a	n/a	n/a
Max pooling	1	2*2	2*2	Valid
Fully connected layer1	250 nodes	n/a	n/a	n/a
Fully connected layer2	150 nodes	n/a	n/a	n/a
Softmax layer	n/a	n/a	n/a	n/a

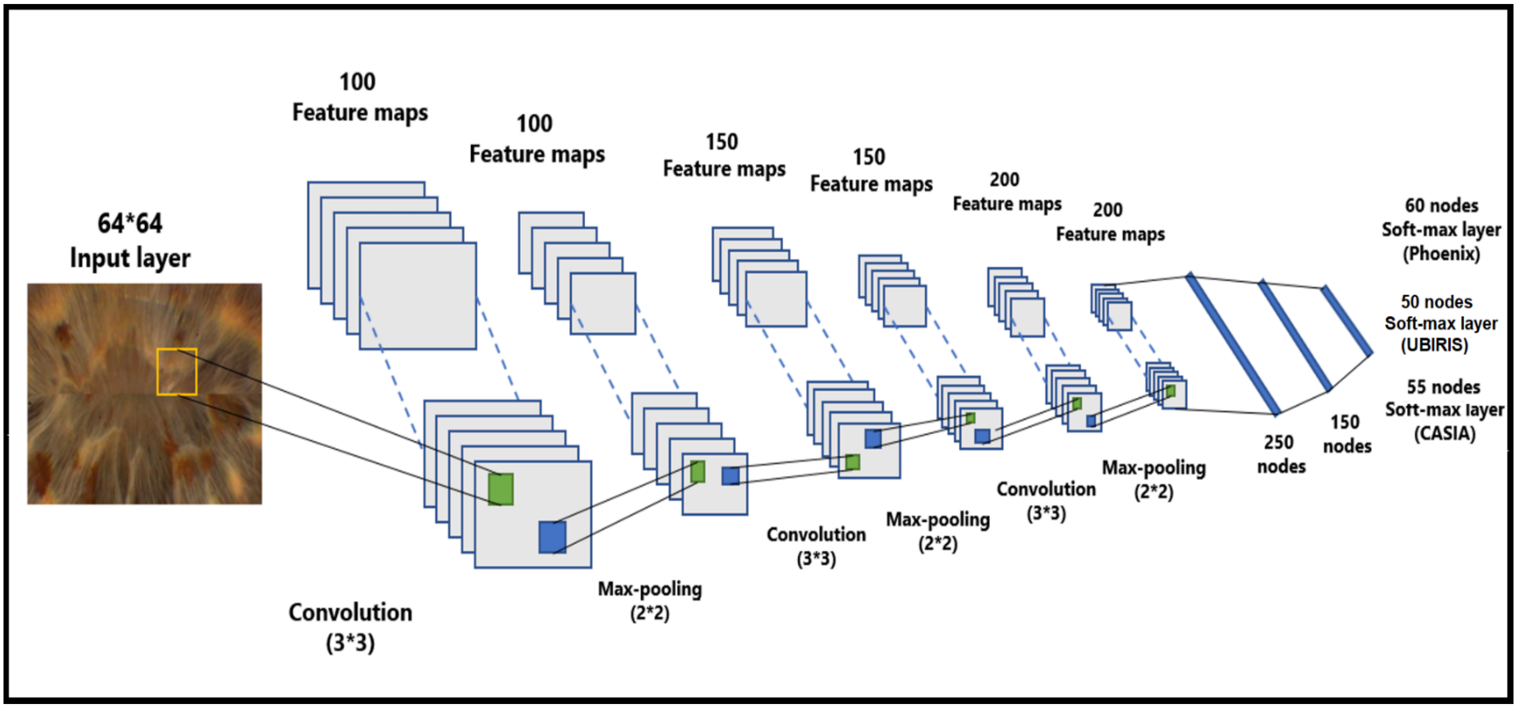


Figure 3.8: The proposed CNN model structure of all datasets.

3.5 Results and discussions

In the next subsections, we will discuss the experimental results related to iris segmentation, iris normalization, recognition, and classification using CNN.

3.5.1 Iris Segmentation and Normalization

In MT, the mask size (n) controls the iris region. In Table 3.3, the accuracy changed according to the value of the (n) parameter. The specular reflect in the pupil is one circular white spot; our algorithm hardly detects a pupil with such environmental nature. The range of (30-45) for mask size (n) in MT achieves the best accuracy. When $(n < 30)$ the final iris mask expanded, including sclera pixels gradually. For $(n > 45)$, the mask loses more information from the iris circle.

TABLE 3.3
THE SEGMENTATION SUCCESS RATE FOR EVERY N PIXEL IN MT.

Mask Size(N) (pixel)	**Success Rate (%)**
60	78.268
55	79.542
50	80.815
45	82.089
40	87.882
35	93.674
30	**99.467**

3.5.2 Recognition and Classification using CNNs

Many trials of experiments were done to arrive at the best tuning of the network parameters of the proposed model architecture previously shown in Table 3.2. Some metrics were measured; to evaluate the performance of the proposed model. The first metric is the recognition accuracy rate (A_{RR}%) which is the portion of correctly iris classifications to the total number of classified irises declared in Equation 3.10. The other metrics are precision, recall, F1-Score, and the training time of the proposed CNN model.

$$\mathrm{A}_{RR}\ (\%) = \frac{\boldsymbol{N_c}}{\boldsymbol{T_c}} * 100 \qquad (3.10)$$

Where ($\boldsymbol{N_c}$) represents the number of correct iris classifications and ($\boldsymbol{T_c}$) represents the total number of classified irises. We divided each data set into two subsets, the first subset is used for training CNN to get the best tuning of the parameters, and the second is used for testing each CNN configuration to get its A_{RR}. In all datasets the ratio between training to testing datasets (R_{test}^{train}) is shown in Equation 3.11.

$$R_{Test}^{Train} = \frac{|Train|}{|Test|} = \begin{cases} 4:1 & in\ case\ of\ CASIA\ and\ UBIRIS \\ 2:1 & in\ case\ of\ Phoenix \end{cases} \qquad (3.11)$$

Where $|Train|$ and $|Test|$ are the cardinality of training and testing subsets, respectively.

During experiments, different CNN architectures differ in their configuration of convolutional layers, max-pooling layers, RELU layers, kernel sizes, strides, and their fully connected layers. Different training parameters [80] like learning rate, number of epochs, patch sizes, and dimensions of input iris images were used. Tables 3.4, 3.5, 3.6, and 3.7 show a part of the experiments done on the model until reaching the best architecture that gives the highest recognition rate. The best architecture for all datasets together is shown in Table 3.6 and its configuration was previously described in Table 3.2. We did not any experiments on UBIRIS. V1 or CASIA V4 with dimensions of (128*128) as shown in Table 3.7 because the dimensions of the iris template are less than this, so we avoided generating extra features and depending on them.

With the Phoenix dataset, the model was trained with 120 iris images for 60 classes of data in the case of each iris left and right and tested against 60 iris images. The model correctly classified all iris images for both left and right irises with an accuracy of recognition rate of 100% for left and right iris. With the UBIRIS. V1 dataset, the model was trained with 200 iris images for 50 classes of data and tested against 50 iris images. The model correctly classified all iris images except only one with an accuracy of recognition rate of 98%. With the CASIA V4 interval dataset, we trained our model with 265 iris images for 55 classes of data in the case of each iris left and right and tested it against 66 iris images. The model correctly classified 65 iris images with the left eye and 66 images with the right eye. The accuracy of the recognition rate of 98.48% and 100% for the left and right iris, respectively. So, the model has an accuracy of 99.24% of the overall CASIA V4-interval dataset. So, the overall accuracy of the proposed model is 99.33%.

TABLE 3.4

ACCURACY OF RECOGNITION RATES OBTAINED FOR DIFFERENT CNN ARCHITECTURES USING THE INPUT IMAGE SIZE OF (256 × 64) PIXELS

	Phoenix		**CASIA V4**		**UBIRIS (%)**
Configuration	**Left (%)**	**Right (%)**	**Right (%)**	**Left (%)**	
[20 80 120] *	93.33	90	96.96	96.96	94
[5 50 100]	93.33	8166	93.93	93.93	96
[5 50 120]	91.66	90	98.48	95.45	94
[5 40 120]	90	88.33	89.39	86.36	92
[120 120 120]	85	88.33	98.48	96.96	96
[20 70 160]	91.66	90	95.45	92.42	90
[120 100 80]	88.33	86.66	93.93	87.87	96
[120 80 50]	90	90	100	95.45	96
[20 80 140 256]	81.66	88.33	80.30	83.33	92
[5 50 100 150]	85	93.33	83.33	84.84	90
[10 80 120 180]	91.66	93.33	92.42	86.36	94
[20 70 160 200]	88.33	78.33	89.39	87.87	94

****Where in the pattern of [x1 x2 x3]: x1, x2, and x3 indicate the number of kernels in each convolutional layer.***

TABLE 3.5

ACCURACY OF RECOGNITION RATES OBTAINED FOR DIFFERENT CNN ARCHITECTURES USING THE INPUT IMAGE SIZE OF (128 × 64) PIXELS

	Phoenix		**CASIA V4**		**UBIRIS (%)**
Configuration	**Left (%)**	**Right (%)**	**Right (%)**	**Left (%)**	
[6 50 150]	93.33	98.33	95.45	93.93	94
[100 150 200]	91.66	90	98.48	90.90	92
[10 50 250]	95	91.66	95.45	93.93	94
[10 100 200]	93.33	88.33	96.96	90.90	96
[120 120 120]	96.66	85	95.45	92.42	94
[100 200 300]	93.33	86.66	95.45	90.90	90
[20 70 160]	96.66	98.33	98.48	93.93	96
[120 80 50]	93.33	93.33	93.93	93.93	96
[10 40 80]	96.66	93.33	93.93	92.42	92
[10 50 100 200]	90.00	91.66	92.42	90.90	94
[50 100 150 250]	90.00	90	89.39	87.87	94
[100 150 200 250]	93.33	91.66	92.42	92.42	92

TABLE 3.6

Accuracy of recognition rates obtained for different CNN architectures using the input image size of (64 × 64) pixels

	Phoenix		CASIA V4		UBIRIS (%)
Configuration	**Left (%)**	**Right (%)**	**Right (%)**	**Left (%)**	
[10 50 100]	95	96.66	98.48	98.48	98
[10 40 80]	98.33	98.33	96.96	98.48	96
[10 100 150]	98.33	95	95.45	96.96	92
[20 70 160]	96.33	95	98.48	95.45	96
[120 120 120]	95	95	96.96	98.48	94
[100 150 200]	**100**	**100**	**98.48**	**100**	**98**
[20 60 120 180]	90	96.66	92.42	95.45	96
[10 50 100 150]	93.33	95	95.45	92.42	92
[50 100 150 200]	95	96.66	96.96	93.93	94
[100 150 200 250]	95	95	95.45	92.42	98
[10 50 100]	95	96.66	98.48	98.48	96
[10 40 80]	98.33	98.33	96.96	98.48	94

TABLE 3.7

Accuracy of recognition rates obtained for different CNN architectures using the input image size of (128 × 128) pixels for Phoenix dataset only

Configuration	**Phoenix left eye (%)**	**Phoenix right eye (%)**
[20 80 120]	83.33	90
[5 50 100]	95	93.33
[5 50 120]	86.66	91.66
[5 40 120]	88.33	96.66
[120 80 50]	93.33	88.33
[5 50 100 150]	93.33	88.33
[10 80 120 180]	95	90
[20 70 160 200]	90	90

Because our problem is a multi-class classification problem, so we have a Precision, recall, and F1-Score measure for each class of the clients of the system as shown in Equations 3.12, 3.13, and 3.14.

$$Precision\ (Class =\ x) = \frac{TP(class = x)}{TP(class = x) + FP(class = x)} \tag{3.12}$$

$$Recall\ (Class =\ x) = \frac{TP(class = x)}{TP(class = x) + FN(class = x)} \tag{3.13}$$

$$F1 - Score\ (Class =\ x) = \ 2 * \frac{Precision(class = x) * Recall(class = x)}{Precision(class = x) + Recall(class = x)} \tag{3.14}$$

Where TP, FP, FN, and x stands for true positive, false positive, false negative, and the number of client's class, respectively. All of the 60 classes of the Phoenix dataset, for left and right iris, have 1, 0.0163, and 0.032 for precision, recall, and F1-Score, respectively. With the UBIRIS. V1 dataset, 49 class has 1, 0.02, and 0.039 for precision, recall, and F1-Score, respectively.

Figure 3.9 shows the average accuracy curves for training for visible light vision datasets. Figure 3.10 shows heatmap representation of activation functions and saliency map representation for all used datasets.

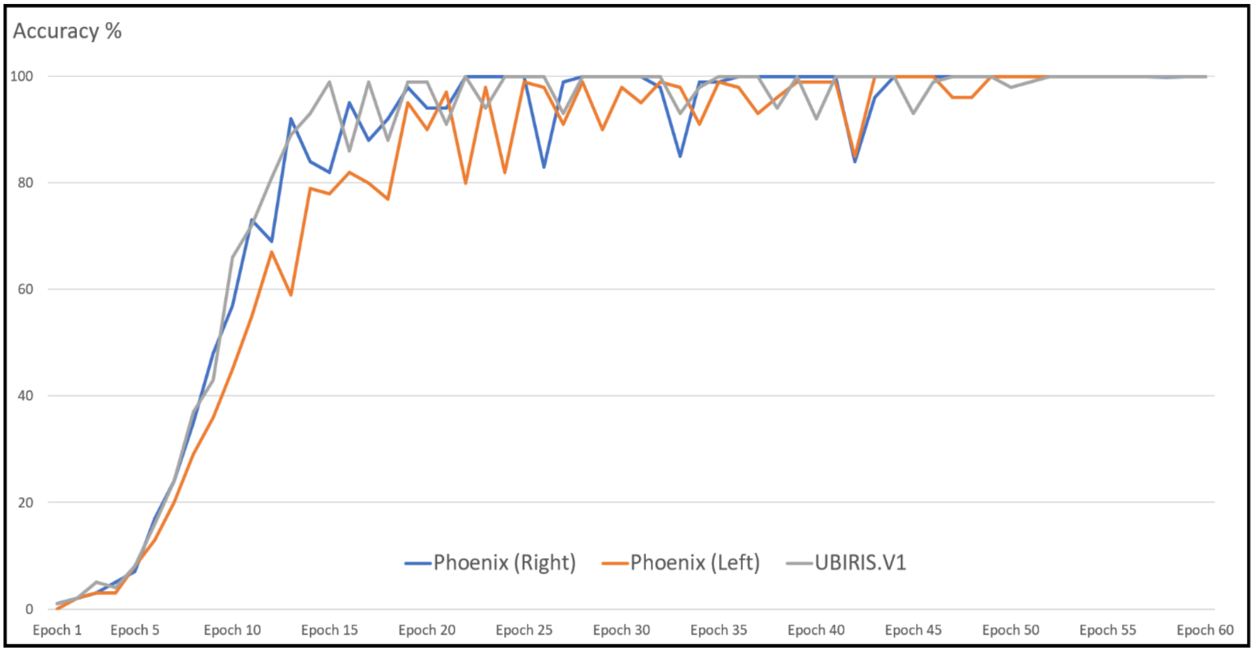


Figure 3.9: Average accuracy curve for the training of datasets.

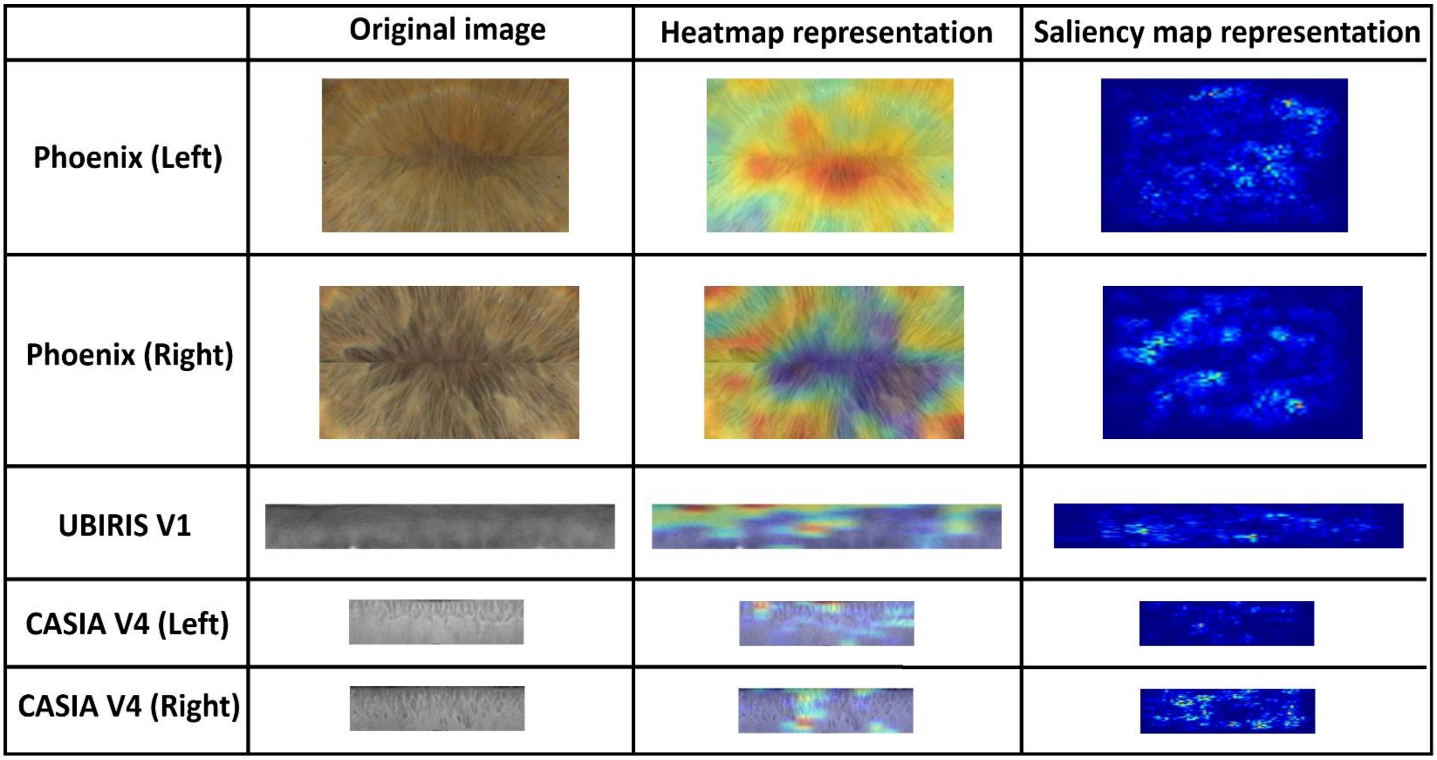


Figure 3.10: heatmap representation of activation functions and saliency map representation for all used datasets.

To deal with the lack of dataset images per subject during training, a simple data augmentation operation was done. This augmentation was based on simple image processing operations on iris images like concatenating some parts from the top, bottom, left, or right of iris images. These new images were added as a part of the training set.

With each of the experiments in the above tables, the number of epochs, batch size, and learning rate was varying until reaching the best values. It is found that a suitable number of epochs needed for training which gives us the highest recognition rate was 60 epochs with a batch size of 40, a learning rate of 0.001 with categorical cross-entropy loss function. Concerning training time, with the final configuration of the CNN model, the training time of the Phoenix dataset was about 17 minutes for each right and left sub-sets, 22 minutes for UBIRIS. V1 dataset and the training time of the CASIA V4 interval dataset was about 25 minutes for each right and left iris sub-sets. It is considered relatively low training time comparing with others, like [15], who addressed the use of deep learning in iris recognition with training time exceeds 6 hours.

3.5.3 Comparative Study

Table 3.8 shows that the proposed CNNs model has competitive results compared to state-of-the-art methods in terms of recognition accuracy.

TABLE 3.8
COMPARISON OF THE PROPOSED SYSTEM WITH OTHER WORKS

Approach	Dataset	Feature extraction	Classification	Recognition accuracy %
Ghanapriya Singha et al. (2020) [74]	UBIRIS.V2	Integer Wavelet Transform (IWT)	Normalized Hamming distance	98.9
Ruqaiya Khanam et al. (2019) [73]	CASIA-Iris-V1	Haar wavelet and Daubechies wavelet	feedforward neural network	94.76
Peter Peer et al. (2019) [67]	CASIA Thousand	Pre-trained Xception	Pre-trained DeepLabV3+ with MobileNet	97.46
HK Rana et al. (2019) [70]	CASIA-Iris-V4	PCA and DWT	SVM	95.40
Maram and Lamiaa (2018) [17]	CASIA-Iris-Interval	pre-trained Alex-Net model	Multi-Class SVM	89
Alaa Al-Waisy et al. (2018) [24]	CASIA-Iris-V3	Convolutional Neural Network	Softmax classifier + fusion	100
Minaee et al. (2016) [16]	CASIA-Iris-Thousand	pre-trained VGG-Net	Multi-Class SVM	90
Saminathan et al. (2015) [75]	CASIA-Iris-V3-interval	Intensity image	Least square method of quadratic SVM	98.50
S. S. Dhage et al. (2015) [69]	Phoenix	Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT)	Euclidean distance	88.50
Bharath et al. (2014) [72]	CASIA-Iris-V3	Radon transform and gradient-based isolation	Euclidean distance	84.17
Sundaram et al. (2011) [71]	UBIRIS. V1	Haralick features	Probabilistic Neural Networks (PNNs)	97
The proposed model	CASIA-Iris-V4	Convolutional Neural Network	Softmax classifier	99.24
	Phoenix			100
	UBIRIS.V1			98

3.6 Chapter Summary

In this chapter, we discussed the basic components of the proposed iris recognition system in general without applying them in IoT environments. A full description of the used datasets is given. We explained the methodologies used for iris segmentation and the masking technique for iris normalization and we discussed the related results of these methodologies in detail. Finally, based on the conducted experiments, we proposed an iris recognition system based on CNN as a feature extractor and a softmax classifier and showed its architecture and accuracy results.

Chapter 4

IoT based Iris Recognition System

In chapter 3, an efficient iris recognition system was discussed in a general way without talking about IoT. In this chapter, we discuss the extension of the proposed iris recognition system, introduced in chapter 3, for IoT environments. We provide a secure method to address the problem of hacking the iris template transmission over the communication channels of IoT systems by protecting the iris using an encryption algorithm based on a chaotic key sequence generated by the sequence of the logistic map and sequence of states of Linear Feedback Shift Register (LFSR). We also discuss the effect of several kinds of noise from different distributions on iris images due to noise interference or bad acquisition, or any other system attacks.

This chapter is organized as follows: Section 4.1 introduces the overall structure of the extended system. Section 4.2 gives a detailed description of the communication paradigm over IoT environments used in experiments. Section 4.3 focuses mainly on the image encryption algorithm used for security enhancement purposes. Section 4.4 focuses on system reliability and evaluation against noised iris images from different probability distributions. Finally, section 4.5 presents the results and discussions.

4.1 The structure of the proposed system

The proposed extended system considers enrollment and authentication processes over the IoT authentication server. The proposed extended iris recognition system follows the classical client-server network communication paradigm, it implements a specific version of chaotic encryption algorithm based on a chaotic key sequence generated by the sequence of the logistic map and sequence of states of Linear Feedback Shift Register (LFSR), respectively. The proposed extended system consists of two sides:

1) client-side
2) and server-side

Client-side is practically implemented using a raspberry pi-2 kit [81], keyboard, mouse, screen, and a cable for internet connection. It is assumed that will be sensing devices to capture the iris images from system users, but datasets play this rule in the proposed work. The server side is practically implemented using a laptop and a cable for internet connection. Figure 4.1 shows the overall structure of the system. Figure 4.2 shows the client and the server sides from testing the proposed system during experiments.

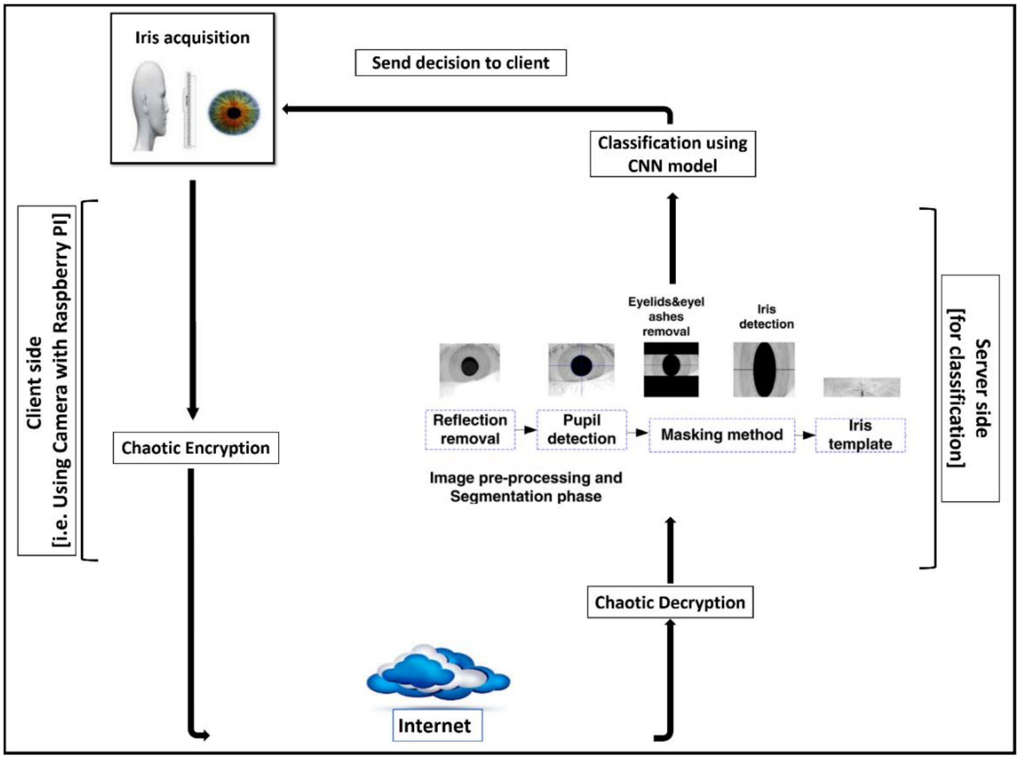
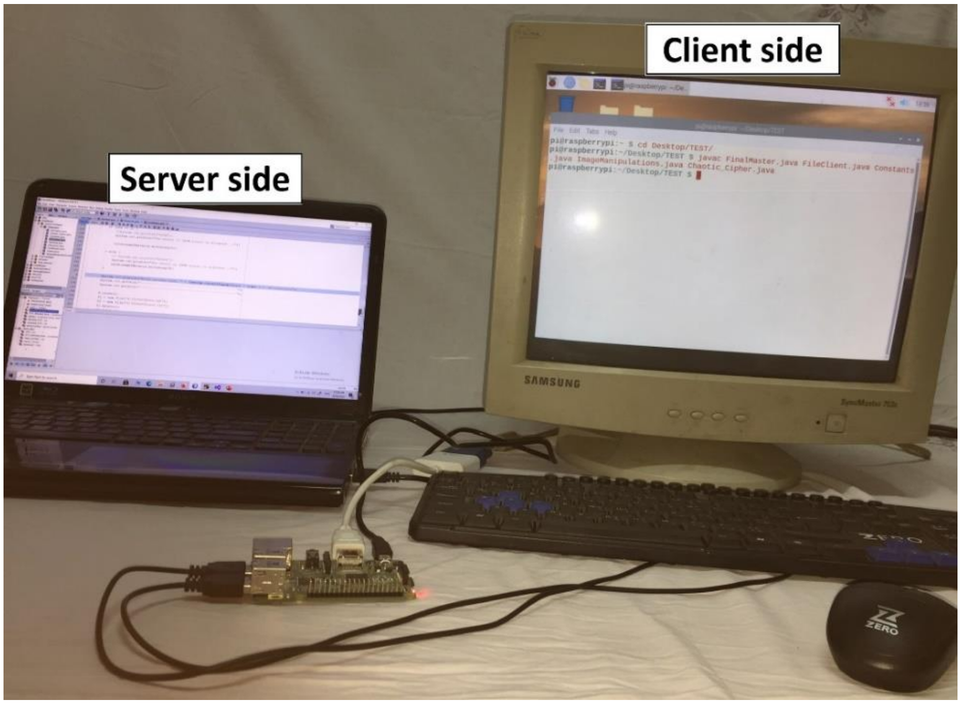

Iris acquisition
Send decision to client
Classification using CNN model
Client side [i.e. Using Camera with Raspberry PI]
Eyelids&eyel ashes removal
Iris detection
Server side [for classification]
Reflection removal
Pupil detection
Masking method
Iris template
Chaotic Encryption
Image pre-processing and Segmentation phase
Chaotic Decryption
Internet

Figure 4.1: The structure of the proposed extended system.



```
pi@raspberrypi:~ $ cd Desktop/TEST/
pi@raspberrypi:~/Desktop/TEST $ javac FinalMaster.java FileClient.java Constants
.java ImageManipulations.java Chaotic_Cipher.java
pi@raspberrypi:~/Desktop/TEST $ 
```

Figure 4.2: The client and server sides from testing the proposed system.

4.2 Communication over IoT

The proposed extended system structure consists of the following key steps at the client-side:

(i) Iris Acquisition

(ii) Iris Encryption

(iii) Sending the encrypted iris to the enrollment and the classification server over a communication channel.

And it consists of the following key steps of the server-side:

(i) Iris Decryption

(ii) Iris segmentation and normalization

(iii) Deep feature extraction using CNNs.

(iv) Classification using a fully connected neural network with a softmax layer.

There are many communication protocols used in the field of IoT systems [82] like message queue telemetry transport (MQTT), the constrained application protocol (CoAP), Hypertext Transfer Protocol (HTTP/2.0), Advanced Message Queuing Protocol (AMQP), an Extensible Messaging and Presence Protocol (XMPP). This research work is based on the classical client-server network communication paradigm [83] as shown in Figure 4.3. The communication steps can be described as follows:

1. The server creates a communication socket.
2. The server binds its socket with any possessed IP address.
3. The server listens to any connection requests from clients.
4. The client creates a communication socket.
5. The client requests a connection with the server.
6. When the server receives a request from a client, it accepts the request and forks a new process that handles that client.
7. A sequence of reading and writing data between them is done according to the needed task.
8. the client ends the connection, by closing its socket.
9. When the client ends the connection, the server kills its forked process that served that client.

The client-side encrypts iris images after the iris acquisition step and sends these encrypted images to the server-side. The server-side then decrypts them, performs all needed iris pre-processing operations of segmentation and normalization. Then, the server-side does feature extraction and classification using the first proposed system, discussed earlier in chapter 3, then sends back the result of classification to the client-side which has two options of accessing or not being able to access the IoT application based only on the classification results.

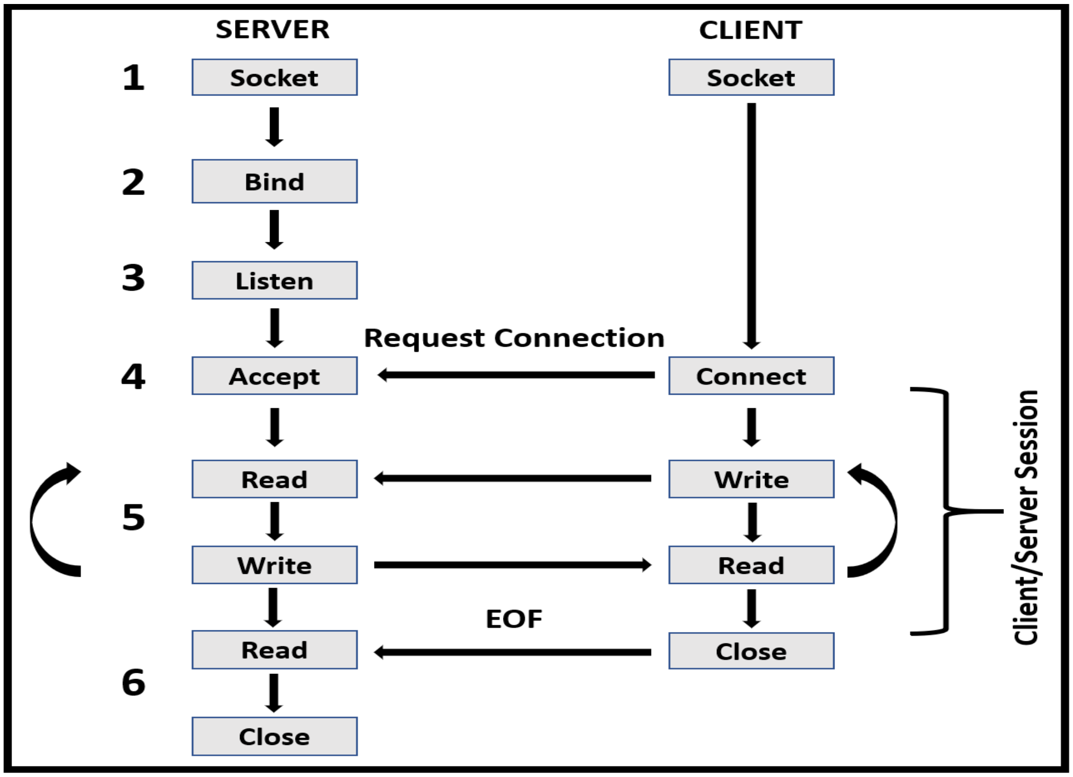


Figure 4.3: The classical client/server model is used by IoT proposed system.

4.3 Image Encryption and Decryption

The encryption is used to increase the proposed system's security level while transmitting iris images over the internet which complicates the way of any system hackers [26]. Iris image encryption of the proposed system is done at the gray level images. The implemented encryption algorithm is based on a chaotic key sequence generated by the sequence of the logistic map and sequence of states of Linear Feedback Shift Register (LFSR) as in [35]. It consists of two main steps:

1- Encryption key sequence (K_Seq) generation
2- Iris image encryption using the generated key sequence.

To generate the encryption key sequence K_Seq; two other non-negative integer finite sequences $K1$and $K2$ of equal lengths, as shown in Equations 4.1 and 4.2, must be generated first.

$$K\_Seq, K1, K2 : \{1,2,3, \dots \dots, p\} \quad \rightarrow \quad [0,255] \qquad (4.1)$$

$$p = w * l \qquad (4.2)$$

Where (w) and (l) are the width and the length of the iris image, respectively because the length of these sequences must equal the length of the iris image that will be encrypted. Terms of $K1$ sequence are generated first by the logistic map Equation 4.3. The bifurcation diagram [84] for the logistic map is shown in figure 4.4.

$$X_{n+1} = r * X_n * (1 - X_n) \quad (4.3)$$

Where (r) is a parameter in the range of the closed interval [2,4] and X_{n+1} and X_n are generic terms in the range of [0,1].

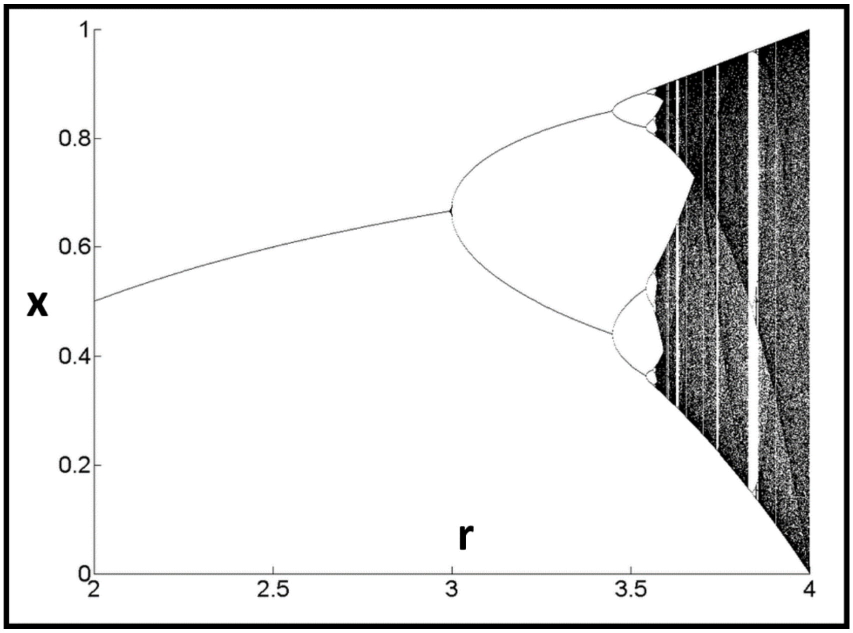


Figure 4.4: Bifurcation diagram for Logistic map[84].

With high values of (r) like $(r = 3.99)$, K_1 will be a chaotic and unexpected sequence. then we round all terms of the obtained sequence by multiplying it by 255 to make sequence values in the range of gray level.

$K2$ sequence is generated by a sequence of states of an 8-bit Linear Feedback Shift Register [85] shown in figure 4.5. This sequence is defined inductively by the recurrence relation [86] in Equation 4.4 with an initial term $K2_1$, called the seed value, equals an integer in the range of [0,255], then perform XOR binary operation on bits of that seed, shift left it with the output of XOR operation and the result will be the second element of the $K2$ sequence and other sequence elements will be generated inductively in the same way.

$$K2_{n+1} = K2_n \gg \left(\bigoplus K2_n\right) \qquad \forall n(\,1 \le n \le p-1) \qquad (4.4)$$

Where $\gg (x)$ denotes shift left operation with the value of x bit, and $\oplus$ denotes the XORing operation of all bits of a term of a sequence.

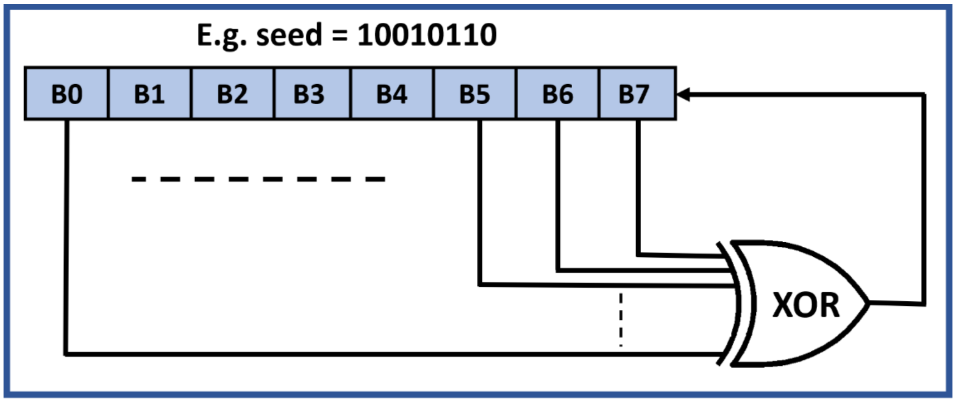


Figure 4.5: Linear feedback shift register [85].

Then K_Seq can be obtained directly from $K1$ and $K2$ sequences by XORing them as shown in Equation 4.5.

$$K\_Seq_n = K1_n \bigoplus K2_n \qquad \forall n(\,1 \le n \le p) \qquad (4.5)$$

Now, after obtaining encryption key sequence K_Seq, encryption of iris image will be done by XORing each pixel of an iris image with its corresponding element in the key sequence as shown in Equation 4.6.

$$IM\_ENC(x,y) = K\_Seq_{x+(y-1)*w} \bigoplus IM\_ORIG(x,y)$$
$$\forall x,y((1 \le x \le w) \wedge (1 \le y \le l)) \qquad (4.6)$$

Where $IM\_ORIG(x,y)$ and $IM\_ENC(x,y)$ denotes the value of the original and encrypted image pixels at (x,y).

The decryption operation is simply the reverse order of this method. As in [35], this LFSR method provides cryptographically better results as compared to the methods that encrypt using a logistic map scheme alone, it provides a high degree of secrecy and security. The original iris image and the encrypted image are highly uncorrelated and perceptually different. For these reasons, our proposed system incorporates this algorithm; to add secure iris transmission for critical IoT applications.

4.4 System Reliability and Evaluation against noised iris images

To ensure the reliability, stability, and robustness of the proposed extended system, we explored how it deals with noised iris data and how the accuracy of the recognition rate of the proposed model would be affected. In general, any iris recognition system will not be embedded in an ideal environment concerning iris acquisition.

So, it is assumed that image acquisition at client sides may gain some external noise due to several reasons like system attacks, environmental dust, interference noise on iris sensing devices, or non-appropriate iris acquisition due to faults of system users, different illumination states …. etc.

The effect of two different kinds of noise is discussed. The first kind is generated randomly from a Gaussian (Normal) distribution, introduced by the French mathematician Abraham DeMoivre in 1733, who used it to approximate probabilities associated with binomial random variables when the binomial parameter n is large, with a mean (μ) [87] zero and several standard deviations (σ) [88]. The probability density function (pdf) and cumulative density function (CDF) of Gaussian (Normal) distribution are shown in Equations 4.7 and 4.8, respectively. Its pdf is shown in figure 4.6.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} * e^{\frac{-(x-\mu)^2}{2*\sigma^2}} \qquad -\infty < x < +\infty \qquad (4.7)$$

$$F(a) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{a} e^{\frac{-(x-\mu)^2}{2*\sigma^2}}\, dx \qquad -\infty < a < +\infty \qquad (4.8)$$

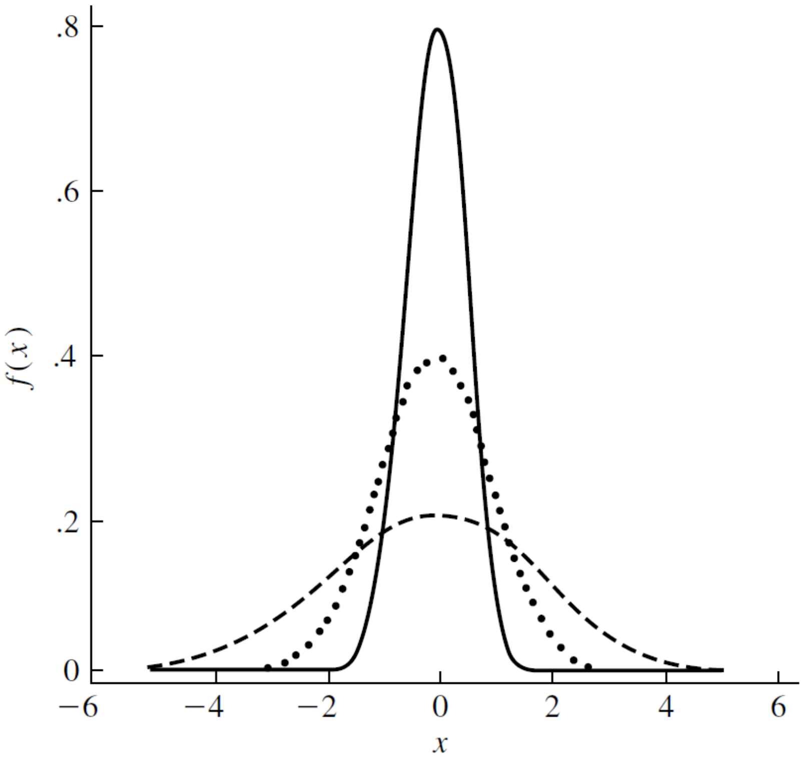


Figure 4.6: The probability density functions for Gaussian (Normal) distribution [88].

The other type of noise is generated by uniform distribution at different intervals [89]. The probability density function (pdf) and cumulative density function (CDF) of the uniform distribution on the interval (α, β) are shown in Equations 4.9 and 4.10, respectively. Its pdf and cdf are shown in figure 4.7, respectively. These randomly generated noise values are added per pixel of iris images.

$$f(x) = \begin{cases} \dfrac{1}{\beta - \alpha} & \alpha < x < \beta \\ 0 & otherwise \end{cases} \tag{4.9}$$

$$F(a) = \begin{cases} 0 & a \leq \alpha \\ \dfrac{a - \alpha}{\beta - \alpha} & \alpha < a < \beta \\ 1 & a \geq \beta \end{cases} \tag{4.10}$$

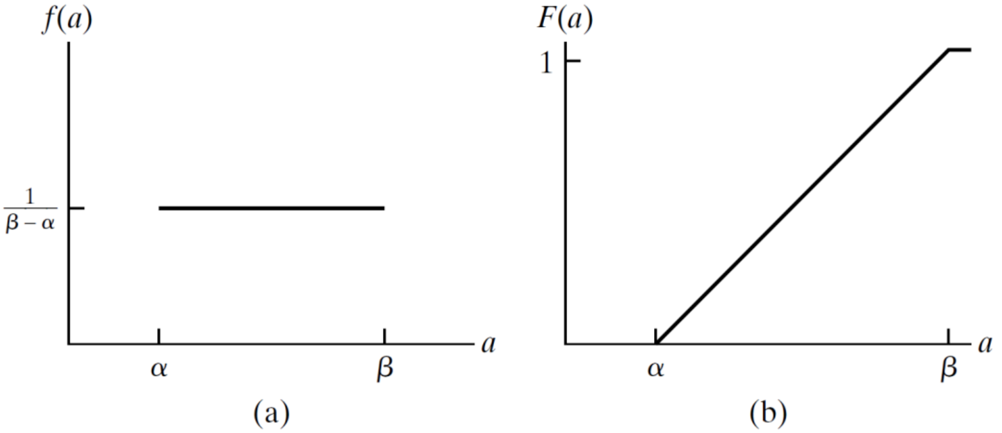


Figure 4.7: (a)The probability density function for the uniform distribution. (b) The cumulative distribution function for the uniform distribution [89].

4.5 Results Analysis and Discussion

In the next subsections, we will discuss the experimental results related to iris encryption, iris decryption, and adding several kinds of noises to iris images.

4.5.1 Image Encryption and Decryption

In all experiments on all dataset images, the values of the parameters needed to generate K_Seq, $K1$, and $K2$ sequences are chosen as shown in assignments Equation 4.10.

$$r = 3.99 \qquad \& \qquad X_1 = 0.1 \qquad \& \qquad K2_1 = (0100101)_2 \qquad (4.10)$$

Figure 4.8 shows the results of the chaotic encryption algorithm implemented in the proposed extended system on all datasets.

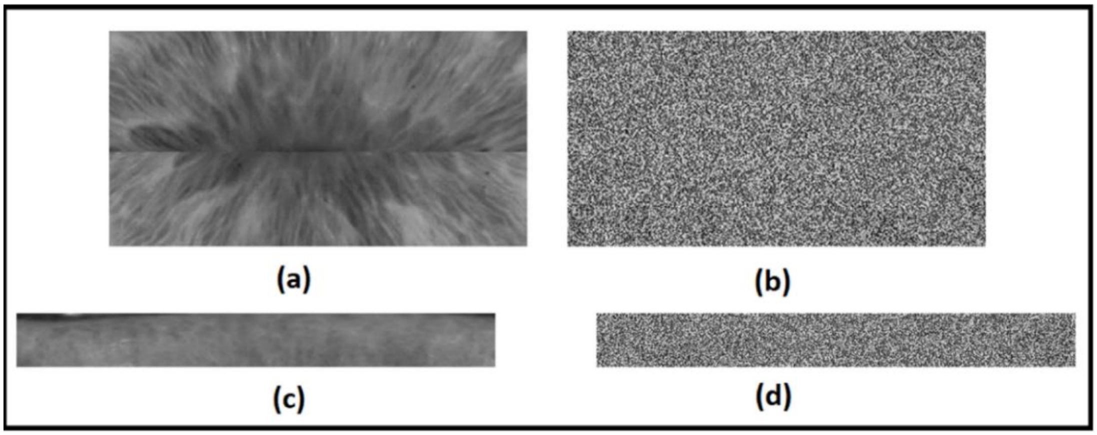


Figure 4.8: Results of Chaotic encryption and decryption. (a) Original image from the phoenix dataset. (b) Encrypted image of the original image from the phoenix dataset. (c) Original image from the CASIA V4 dataset. (d) The Encrypted image of the original image from the CASIA V4 dataset.

4.5.2 Testing the Effect of Noise on The Proposed Extended System

The final CNN configuration that yields the highest recognition rate in the ideal case of iris images without any added noise, discussed earlier in chapter 3, is tested again against the noised version of the testing set. Regards the added noise from Gaussian distribution, Figures 4.9 show the results of adding this kind of noise with different standard deviations. Table 4.1 shows the accuracy of the recognition rate of the proposed extended system concerning noised iris images. Figure 4.10 shows the accuracy of the recognition degradation curve after adding this kind of noise.

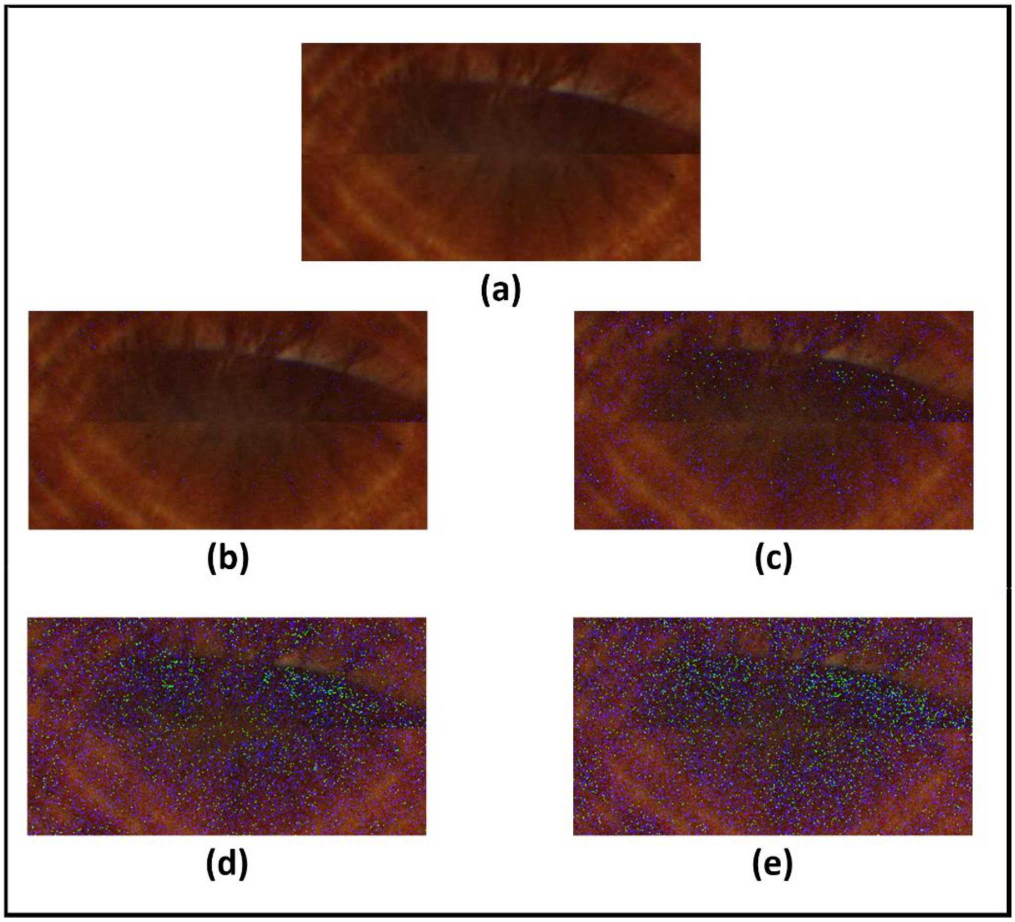


Figure 4.9: Results of adding Gaussian noise with different standard deviations. (a) Original image. (b) Standard deviation = 5 (c) standard deviation = 10 (d) standard deviation = 15 (e) standard deviation = 20

TABLE 4.1
THE MODEL RECOGNITION RATE AGAINST NOISED IRIS IMAGES (GAUSSIAN NOISE)

	UBIRIS	***Phoenix***		**CASIA V4**	
		Left iris	**Right iris**	**Left iris**	**Right iris**
5	98	100	100	100	98.484
10	98	96.666	100	96.969	95.454
15	98	88.333	96.666	92.424	93.939
20	92	80.000	88.333	86.363	87.878

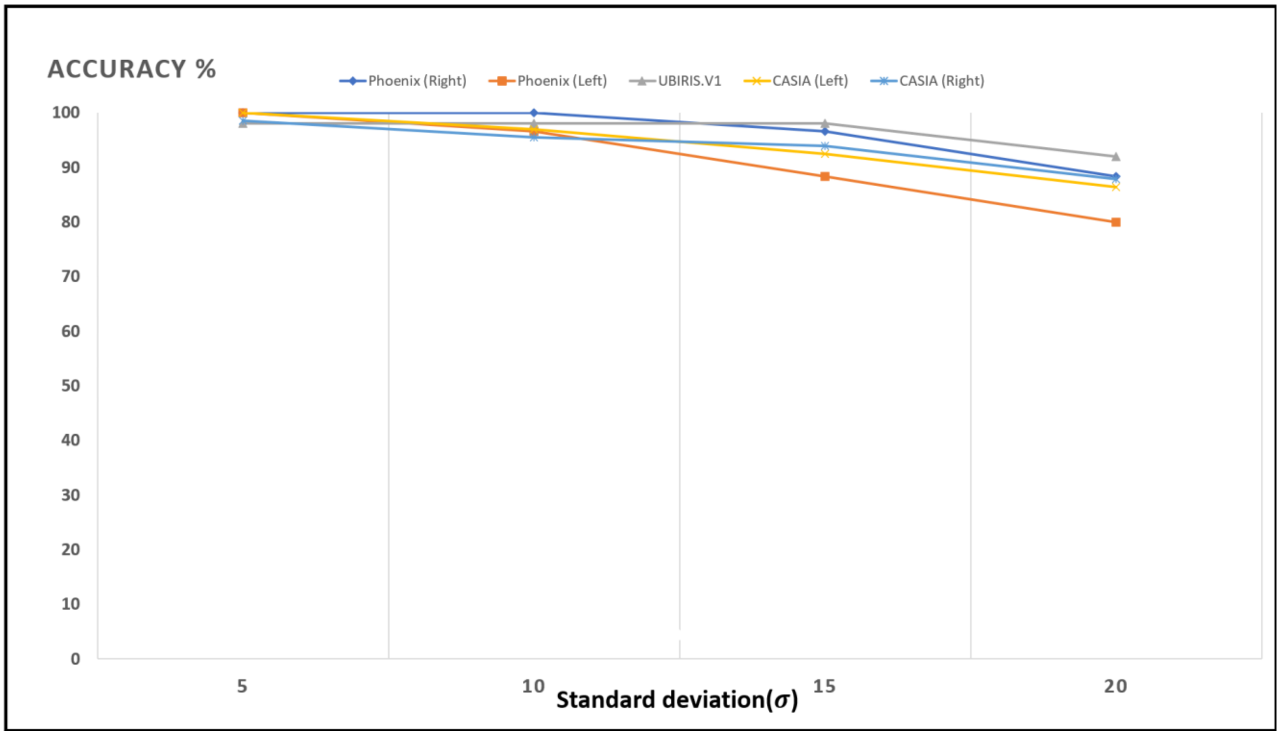


Figure 4.10: Accuracy of recognition degradation curve after adding Gaussian noise with different standard deviations.

These results show how well the proposed model deals with noised iris images from Gaussian (Normal). The proposed extended system shows a low degradation of recognition accuracy rates in the case of using noise from Gaussian distribution with different standard deviations.

Regards the added noise from a uniform distribution, Figure 4.11 shows the results of adding this kind of noise within different intervals. Table 4.2 shows the accuracy of the recognition rate of the proposed extended system concerning noised iris images. Figure

4.12 shows the accuracy of the recognition degradation curve after adding this kind of noise.

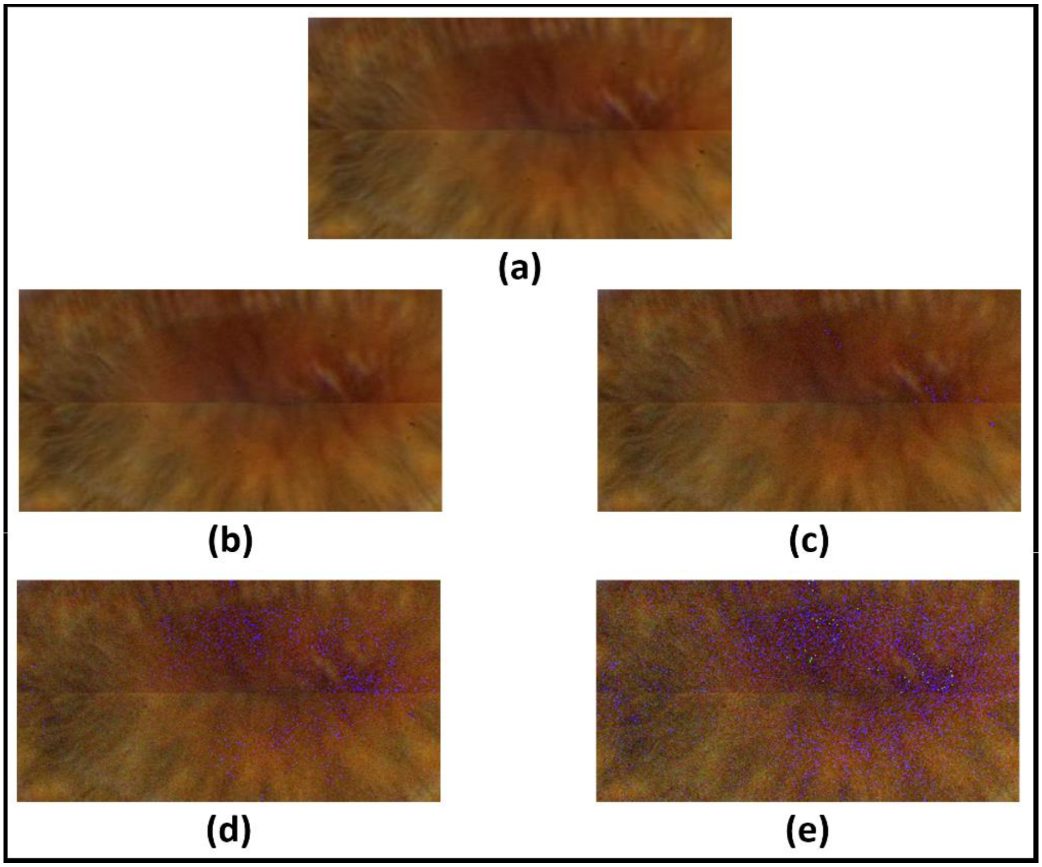


Figure 4.11: Results of adding uniform noise within different intervals. (a) Original image. (b) Within interval of [-5, +5] (c) within interval of [-10, +10] (d) within interval of [-15, +15] (e) within interval of [-20, +20].

TABLE 4.2

THE MODEL RECOGNITION RATE AGAINST NOISED IRIS IMAGES (UNIFORM NOISE)

		Phoenix		**CASIA V4**	
Interval	**UBIRIS**	**Left iris**	**Right iris**	**Left iris**	**Right iris**
$[-5,+5]$	98	100	100	100	98.484
$[-10,+10]$	98	100	100	100	98.484
$[-15,+15]$	96	96.666	96.666	96.969	95.454
$[-20,+20]$	74	83.333	85.000	89.393	92.424

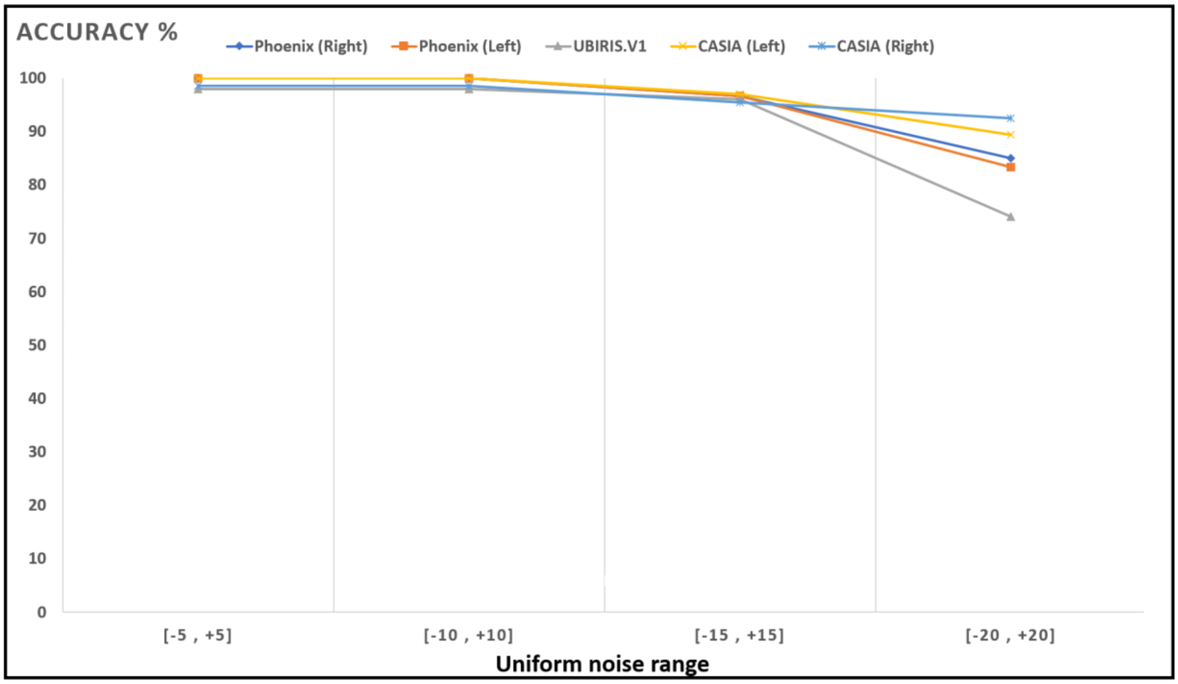


Figure 4.12: Accuracy of recognition degradation curve after adding uniform noise at different intervals.

These results show how well the proposed system deals with noised iris images from the uniform distribution. The proposed extended system shows a low degradation of recognition accuracy rates in the case of using noise from the uniform distribution with different intervals which vary in width.

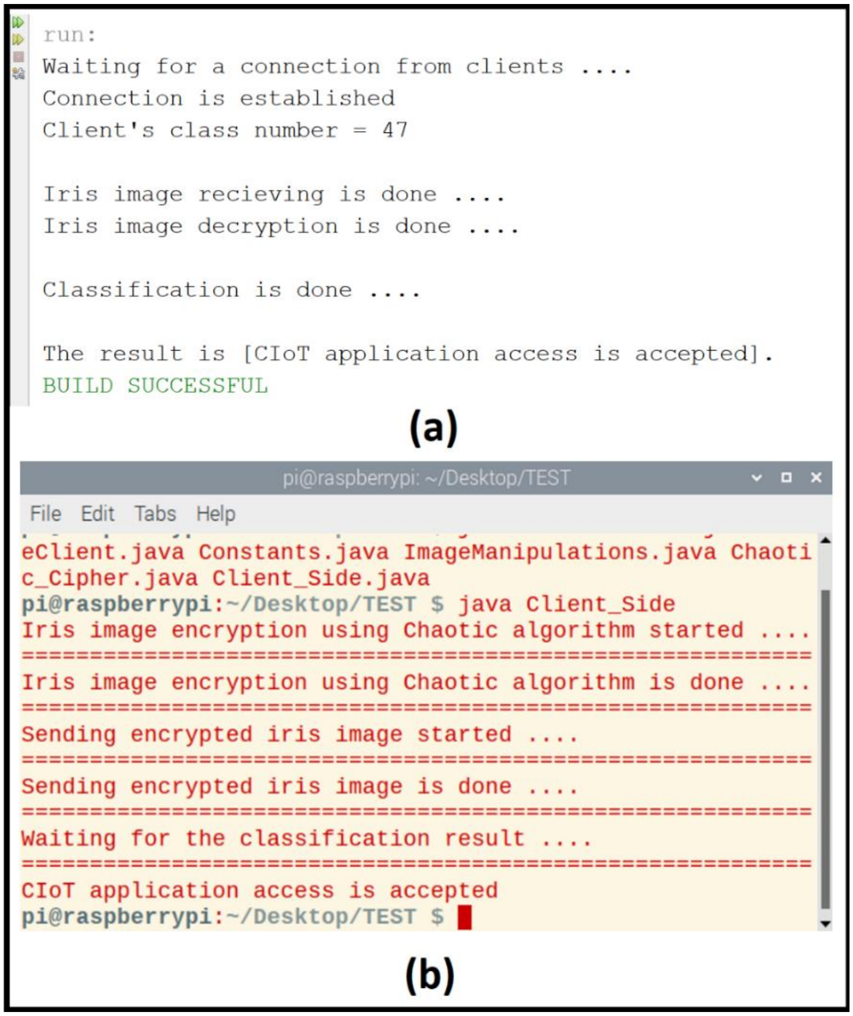
4.5.3 The experimental connection actions sequence of the proposed system

The overall practical steps of the experiments at the server and client sides are shown in figure 4.13 as follows:

1) First, the server-side which is responsible for the classification task is set up for listening to connection requests from clients.
2) When the client requests a connection with the server, the server accepts it.
3) After the client performs iris image acquisition, it performs iris decryption using a chaotic algorithm.
4) The server receives the encrypted image and decrypts it to obtain the original one.

5) Then performs iris pre-processing operations and passes the iris image to the trained classifier.
6) Then send the result of classification to the client whose IoT application access request will be accepted or rejected based on the classification result.

All experiments were done on a lab top, which is used as the server-side of the system, with Core i5 CPU and 6 GB of RAM, and the codes were written in MATLAB, Java, and Python languages. And raspberry pi 2 [81] for the client-side of the system and its scripts are written in Java language.



```
run:
Waiting for a connection from clients ....
Connection is established
Client's class number = 47

Iris image recieving is done ....
Iris image decryption is done ....

Classification is done ....

The result is [CIoT application access is accepted].
BUILD SUCCESSFUL
```

(a)

```
eClient.java Constants.java ImageManipulations.java Chaoti
c_Cipher.java Client_Side.java
pi@raspberrypi:~/Desktop/TEST $ java Client_Side
Iris image encryption using Chaotic algorithm started ....
===========================================================
Iris image encryption using Chaotic algorithm is done ....
===========================================================
Sending encrypted iris image started ....
===========================================================
Sending encrypted iris image is done ....
===========================================================
Waiting for the classification result ....
===========================================================
CIoT application access is accepted
pi@raspberrypi:~/Desktop/TEST $
```

(b)

Figure 4.13: The connection actions sequence (a) Server-side. (b) Client-side.

4.6 Chapter Summary

In this chapter, we discussed the extension of the proposed system in a generic IoT environment. The classical client/server model, which is used as the communication methodology in the extended system, is shown. We discussed iris encryption and decryption over communication channels using an algorithm based on a chaotic key sequence generated by the sequence of the logistic map and sequence of states of Linear Feedback Shift Register. Finally, we studied the effect of adding different kinds of noise (Gaussian and Normal) to iris images to evaluate the reliability of the proposed system.

Chapter 5

An Iris Recognition System for ATMs and Banking systems

Nowadays, the identity verification of banks' clients at Automatic Teller Machines (ATMs) is a very critical task. Clients' money, data, and crucial information need to be highly protected. The classical ATM verification method using a combination of credit card and password has a lot of drawbacks like Burglary, robbery, expiration, and even sudden loss. Recently, iris-based security plays a vital role in the success of the IoT-based security framework. The iris biometric, as discussed earlier, eliminates a lot of security issues, especially in smart IoT-based applications, principally ATMs.

In this chapter, we give a concrete IoT environment, which is the ATMs and banking systems, and show the integration of the proposed extended iris recognition system in this environment. This chapter is organized as follows: Section 5.1 gives a brief overview of Electronic banking (E-banking). Section 5.2 describes the overall structure of the extended system for ATMs and banking systems. Section 5.3 shows the experimental connection action sequence of the proposed system.

5.1 E-banking

Electronic banking (E-banking) was born as a result of globalization, competition, and the rapid growth of IT systems. It has become the self-service delivery channel that allows banks to provide information and offer services to their customers with more convenience via several technology services like the Internet and mobile phone [90]. Therefore, several benefits are offered; convenience, ease to use, low cost, the time factor, fast delivery, and online bill payment. Electronic banking is an inexpensive way to conduct banking business, exchange information, and buy and sell goods or services from any place at any time. Also, it is a way to keep the existing customers and attract others to the bank.

E-banking offers several benefits for both banks and customers. On the one hand, for banks, e-banking allows minimizing operational costs through the reduction of physical facilities, staffing resources required, and waiting times in branches increasing sales performance. On the other hand, for the customers, e-banking enables them to access account information and perform banking transactions electronically anytime and anywhere. Besides, the use of e-banking allows saving time since the customer does not have to be physically present at the bank's local [91].

Despite the many benefits provided by e-banking, the increase of distance between banks and customers may lead to security concerns and a lack of confidence. The number of attacks focused on electronic banking system vulnerabilities has been steadily growing during past years [92]. For this, the security and privacy of electronic banking services got the attention of researchers due to their strong influence on business performance and customer satisfaction. Banks that offer electronic access to their banking systems should develop efficient security models which aim to provide authenticated and secured communications through insecure channels [93]. To address some of these issues, this chapter proposes a novel efficient full authentication system for ATMs based on a bank's mobile application and a visible light environments-based iris recognition.

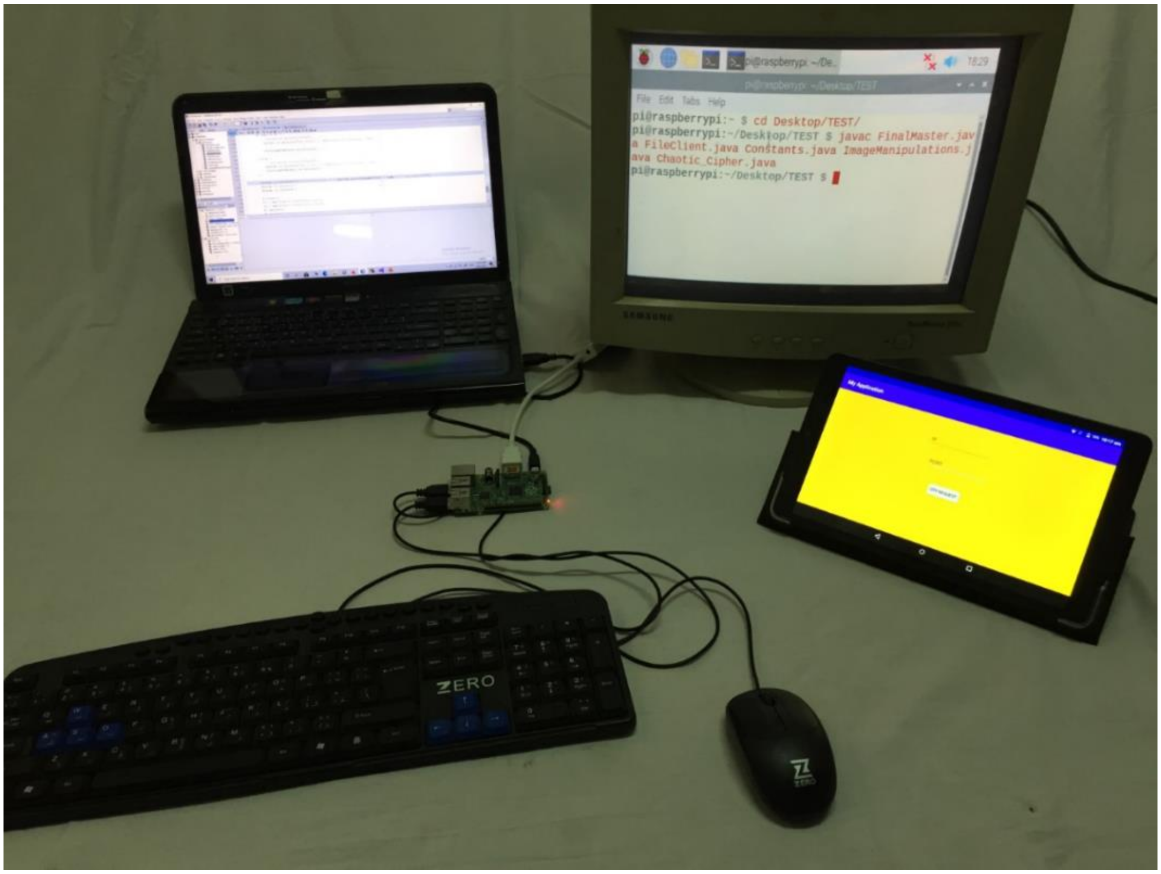
5.2 Proposed System

The proposed system for ATMs and banking systems considers enrollment and authentication processes over the IoT authentication bank server. The proposed system consists of two main sides:

(i) The client-side is practically implemented by mobile phone, upon which the bank application exists, and raspberry pi-2 kit [81], with other accessories like keyboard, mouse, and screen, which represent the ATM node.

(ii) The server-side is practically implemented by a laptop. It is assumed that will be sensing devices to capture the iris images from system users.

The physical connection of the proposed system is shown in Figure 5.1.



```
pi@raspberrypi:~ $ cd Desktop/TEST/
pi@raspberrypi:~/Desktop/TEST $ javac FinalMaster.jav
a FileClient.java Constants.java ImageManipulations.j
ava Chaotic_Cipher.java
pi@raspberrypi:~/Desktop/TEST $
```

Figure 5.1: The practical physical connection and used devices in the experimental structure.

As shown in Figure 5.1, the system starts by requesting a One-Time Password (OTP) by clients' mobile applications from the bank's server which replies with an OTP, that valid for two minutes. Then the client enters the acquired OTP to ATM which captures his/her eye images, encrypts them, and sends the OTP after the encryption of eye images to the bank's server.

Then the bank's server decrypts the received eye images. Hence, it performs all the needed pre-processing operations - like segmentation and normalization - to extract the iris templates and classify them. It checks the correctness of received OTPs with the classified person, and finally sends the final decision about accessing ATMs to clients.

The proposed system structure consists of the following key steps of the client-side:

(i) The client's mobile application requests One-Time Password (OTP) from the server,

(ii) Client's eye image acquisition by sensing devices of ATM,

(iii) Client's eye image encryption using a chaotic algorithm, and

(iv) Finally, sending the client's encrypted eye image to the bank's verification server over the internet.

And it consists of the following key steps at the server-side:

(i) Denoting OTPs to clients,

(ii) Client's encrypted eye image decryption,

(iii) Iris segmentation and normalization,

(iv) Extracting iris features using CNNs constructed model,

(v) Client's iris classification using FCNNs with Softmax layer,

(vi) Checking the correctness of received OTP with the classified class OTP,

(vii) Sending final decision about ATM accessing to clients.

The explanatory sequence diagram of the proposed system is shown in Figure 5.3.

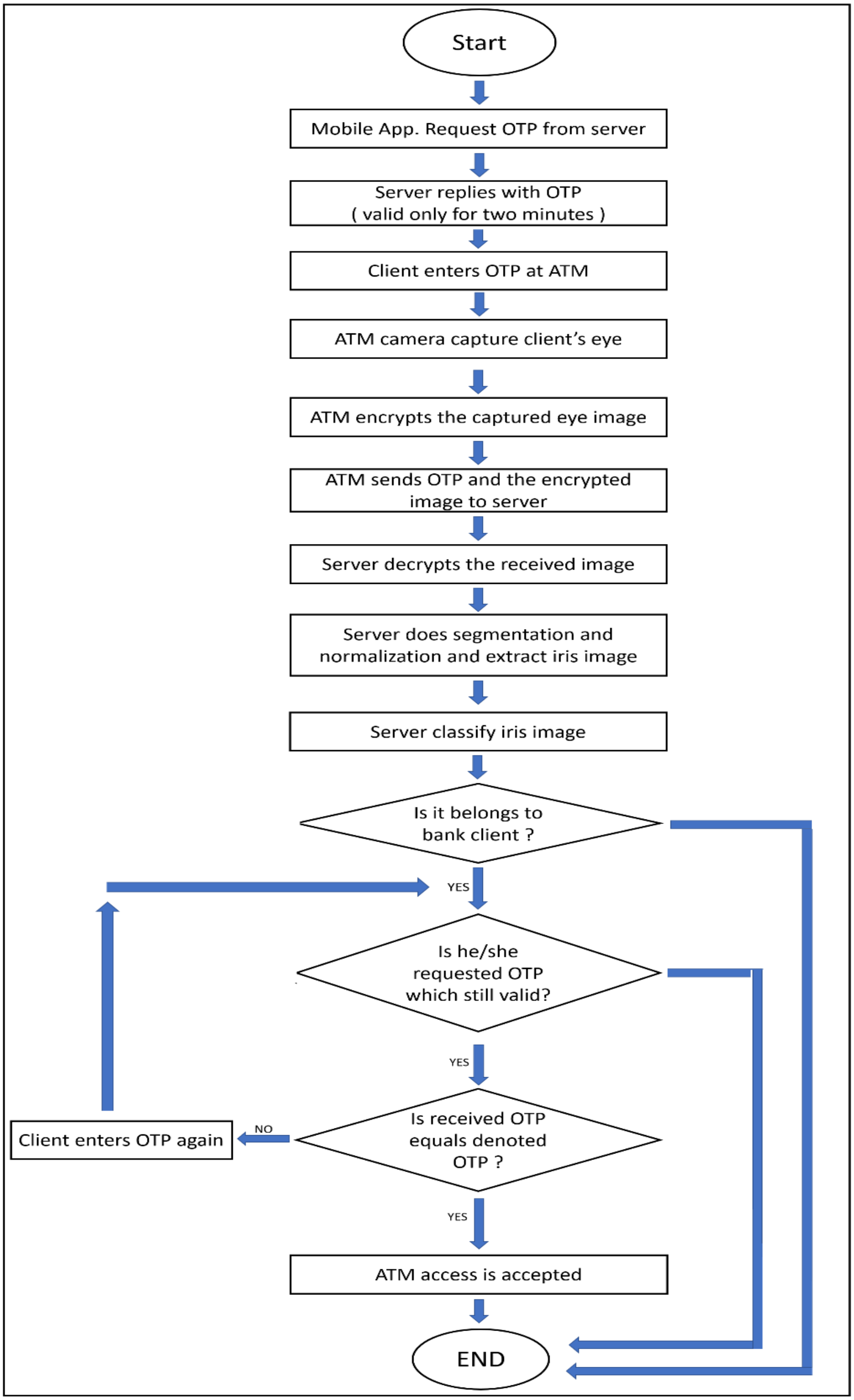

Start
Mobile App. Request OTP from server
Server replies with OTP
(valid only for two minutes)
Client enters OTP at ATM
ATM camera capture client's eye
ATM encrypts the captured eye image
ATM sends OTP and the encrypted image to server
Server decrypts the received image
Server does segmentation and normalization and extract iris image
Server classify iris image
Is it belongs to bank client ?
YES
Is he/she requested OTP which still valid?
YES
Is received OTP equals denoted OTP ?
NO
Client enters OTP again
YES
ATM access is accepted
END

Figure 5.2: The flowchart of the proposed system.

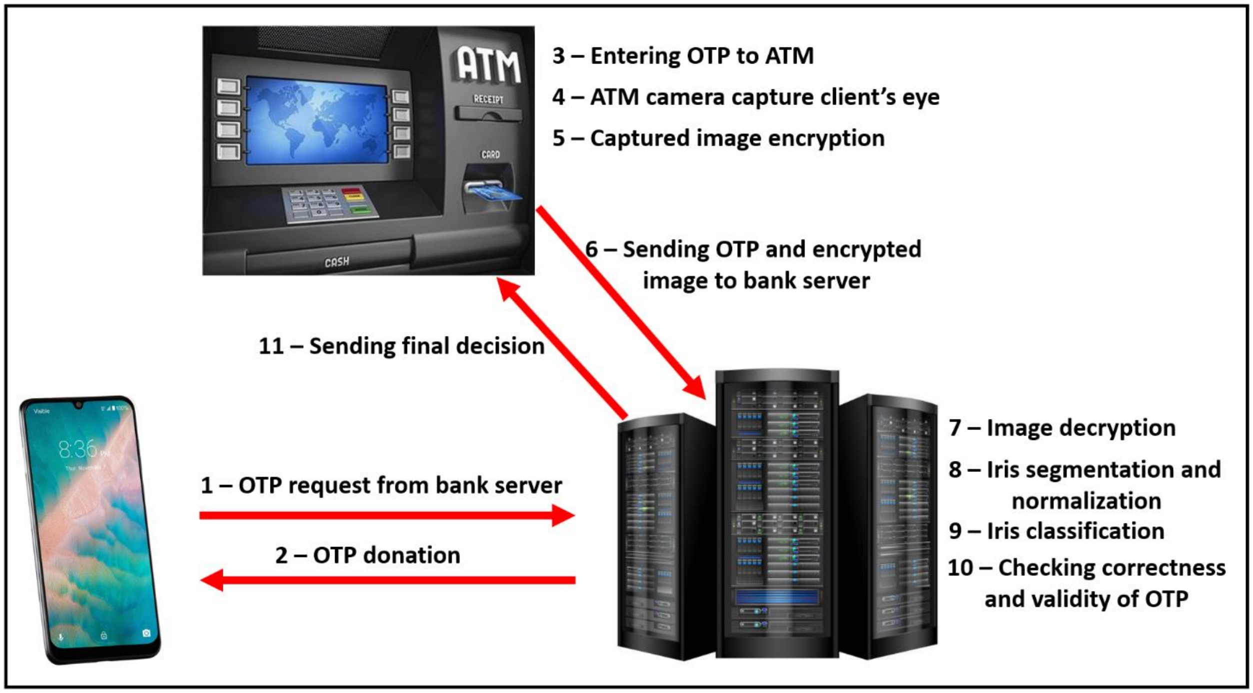

3 – Entering OTP to ATM
4 – ATM camera capture client's eye
5 – Captured image encryption
6 – Sending OTP and encrypted image to bank server
11 – Sending final decision
7 – Image decryption
8 – Iris segmentation and normalization
9 – Iris classification
10 – Checking correctness and validity of OTP
1 – OTP request from bank server
2 – OTP donation

Figure 5.3: The explanatory sequence of the proposed system.

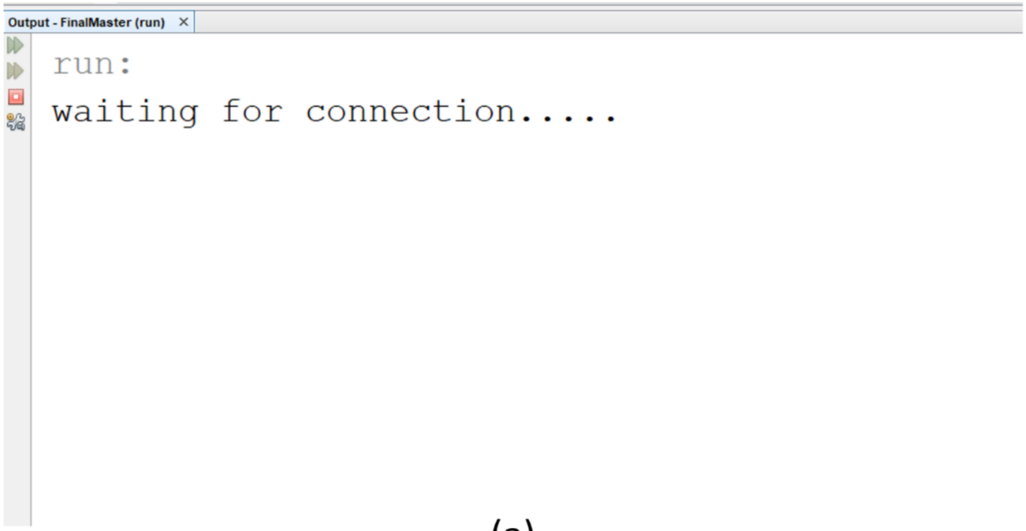
In the proposed system, the communication between the bank server and clients' mobile applications or ATMs over the internet [82] follows the classical client-server networking paradigm [83]. The communication steps concerning the proposed environment are as follows:

1. The bank server creates a communication socket known as a listening socket.
2. Bank server binds its listening socket with any possessed IP address and specific port number which must be known to clients' mobile applications and ATMs.
3. The bank server listens to any connection requests from clients.
4. The client's mobile applications also create communication sockets and request connections to the bank server to receive the OTPs.
5. When the bank server receives a request from a client's mobile application, it accepts the request and forks a new process that handles that client and denotes the required OTP to the client. Then the communication ends, and the forked process will be killed.

6. ATM also creates a communication socket, requests connection to the bank server, and sends the received OTP on mobile application and the eye image after encryption to the bank server.
7. When the bank server receives a request from an ATM, it accepts the request and forks a new process that handles it, performs decryption, segmentation, normalization, classification, and sends a final decision about accessing ATMs. Then the communication ends, and the forked process will be killed.
8. The bank server keeps listening to any coming communication requests from the client's mobile applications or ATMs.

5.3 The experimental connection action sequence of the proposed system

The practical steps of the experiments of the proposed system are shown in the following Figures. First, the bank server task is set up for listening to connection requests from clients' mobile applications or ATMs as shown in Figure 5.4.

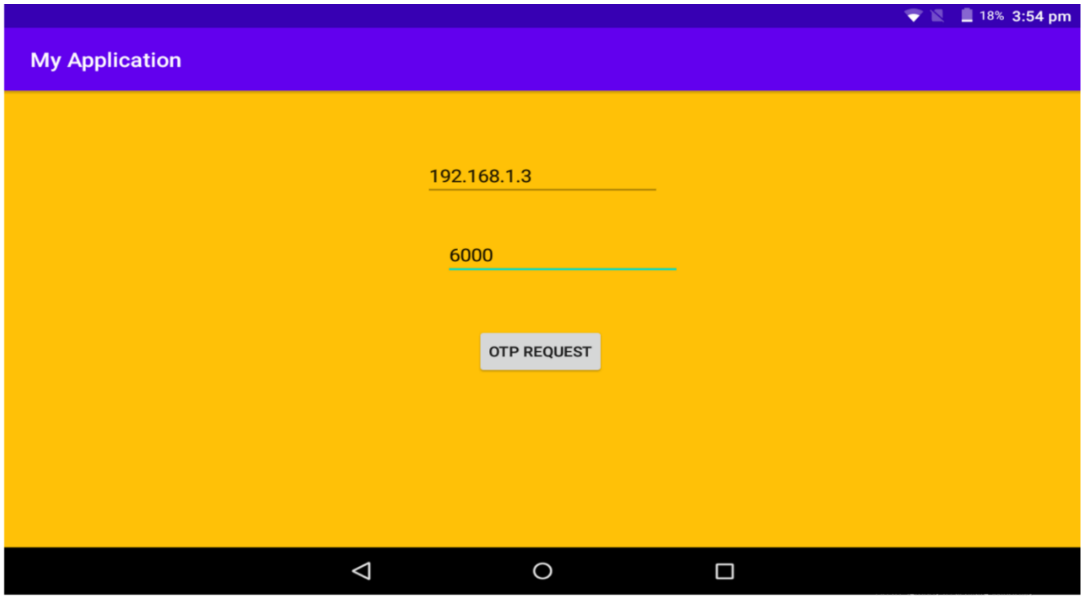


```
run:
waiting for connection.....
```

(a)

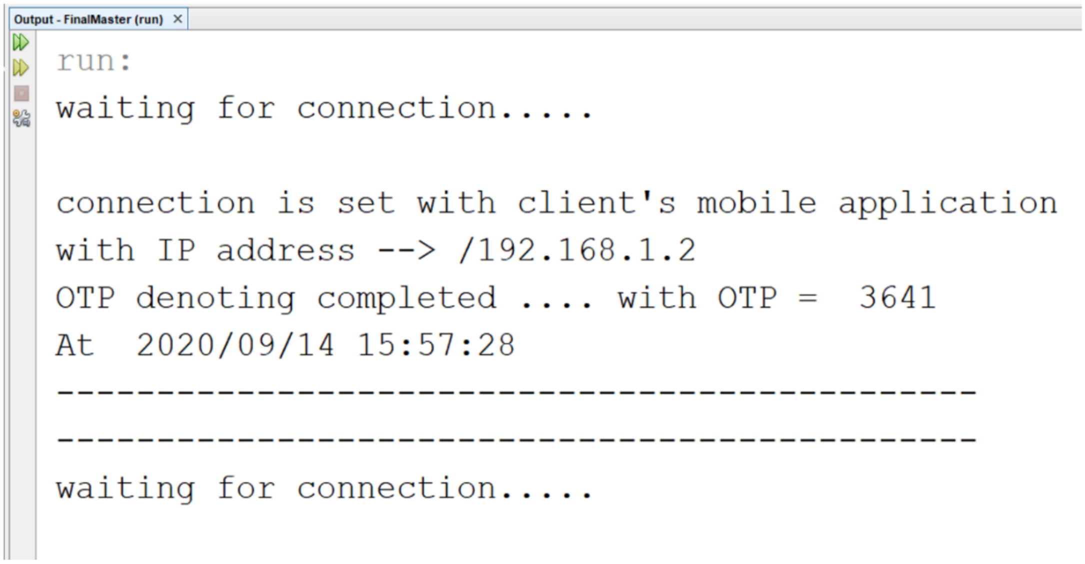
Figure 5.4: Bank server is waiting for connections.

When the client's mobile application requests an OTP, the bank server accepts the request and denotes an OTP to the client as shown in Figures 5.5, 5.6, and 5.7.



(b)

Figure 5.5: Client requests an OTP.

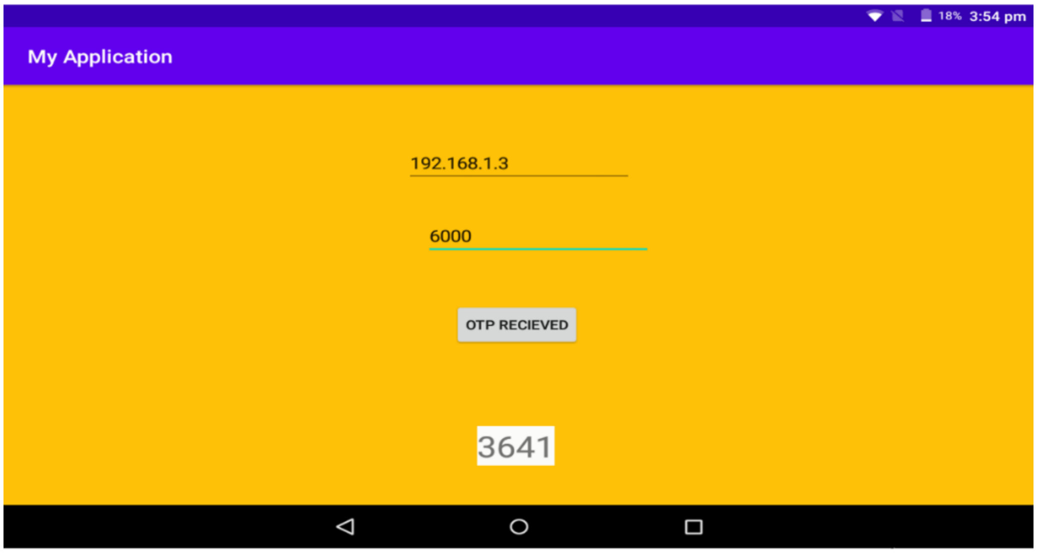


```
run:
waiting for connection.....

connection is set with client's mobile application
with IP address --> /192.168.1.2
OTP denoting completed .... with OTP =  3641
At  2020/09/14 15:57:28
----------------------------------------------
----------------------------------------------
waiting for connection.....
```

(c)

Figure 5.6: OTP donation by bank server.



192.168.1.3

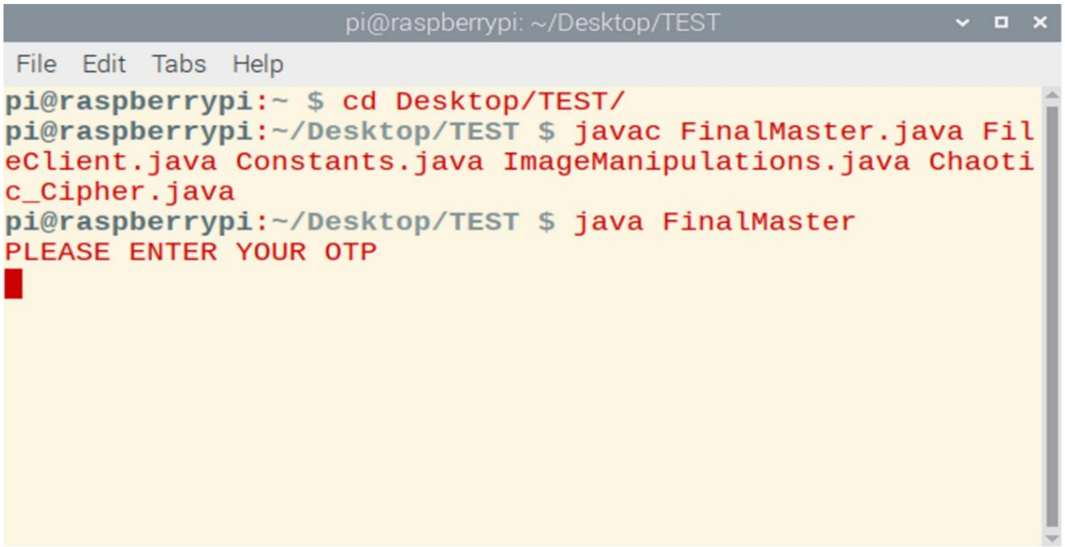
6000

3641

(d)

Figure 5.7: The client receives an OTP.

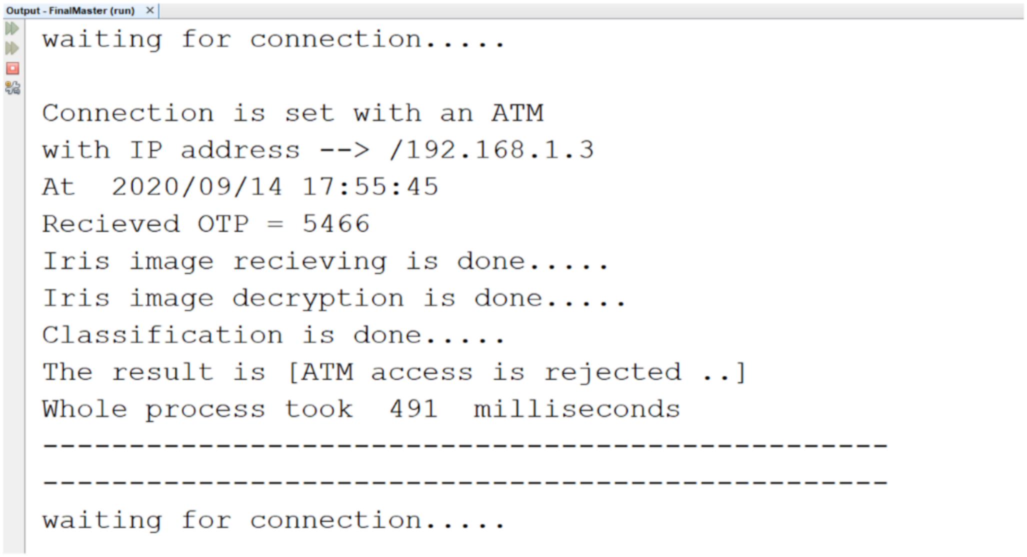
When ATM's client request ATM access by entering the received OTP and capturing his/her eye image, the server receives his/her request, perform all needed operations like segmentation, normalization, classification, and OTP correctness checking as shown in Figures 5.8 and 5.9.



```
pi@raspberrypi:~ $ cd Desktop/TEST/
pi@raspberrypi:~/Desktop/TEST $ javac FinalMaster.java Fil
eClient.java Constants.java ImageManipulations.java Chaoti
c_Cipher.java
pi@raspberrypi:~/Desktop/TEST $ java FinalMaster
PLEASE ENTER YOUR OTP
```

(e)

Figure 5.8: ATM enters received OTP and captures eye image.



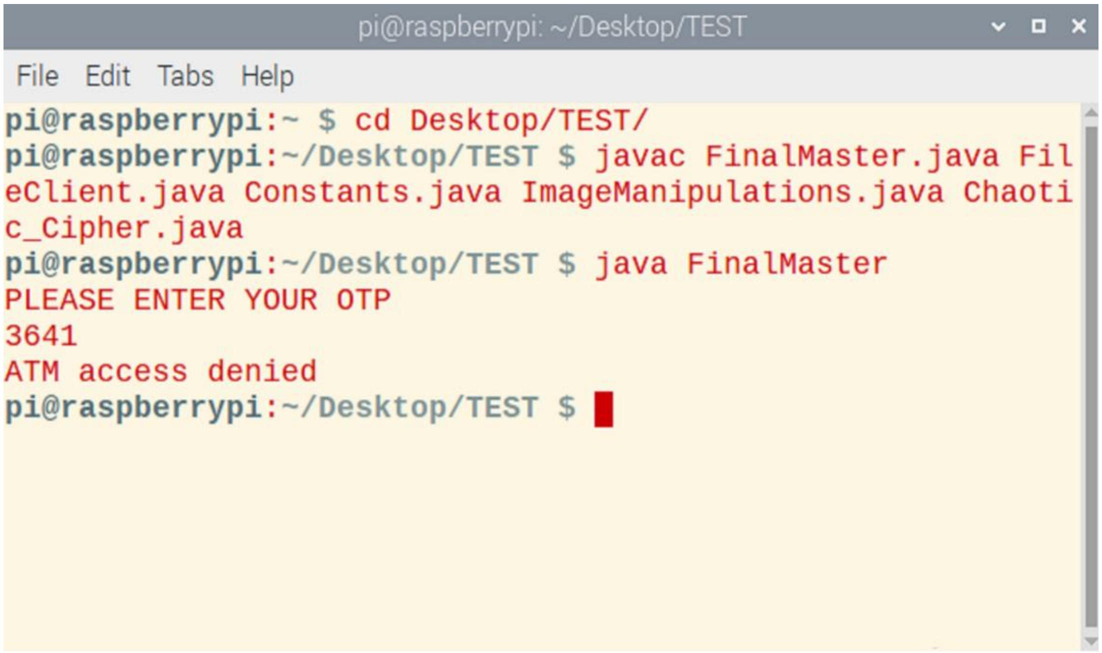
```
waiting for connection.....

Connection is set with an ATM
with IP address --> /192.168.1.3
At  2020/09/14 17:55:45
Recieved OTP = 5466
Iris image recieving is done.....
Iris image decryption is done.....
Classification is done.....
The result is [ATM access is rejected ..]
Whole process took  491  milliseconds
-------------------------------------------------
-------------------------------------------------
waiting for connection.....
```

(f)

Figure 5.9: The bank server determines the final decision.

Then the final access decision is sent to clients by accepting or rejecting ATMs access as shown in Figure 5.10.



```
pi@raspberrypi:~ $ cd Desktop/TEST/
pi@raspberrypi:~/Desktop/TEST $ javac FinalMaster.java Fil
eClient.java Constants.java ImageManipulations.java Chaoti
c_Cipher.java
pi@raspberrypi:~/Desktop/TEST $ java FinalMaster
PLEASE ENTER YOUR OTP
3641
ATM access denied
pi@raspberrypi:~/Desktop/TEST $
```

(g)

Figure 5.10: The client receives the final decision about accessing the ATM.

5.4 Chapter Summary

In this chapter, we discussed the importance of E-banking systems in this era. a real-life scenario from IoT environments, which is ATMs for banking systems, is introduced in detail. We showed how to embed the proposed iris recognition system, discussed in chapter 4, in this application.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, an efficient iris recognition system based on chaotic encryption and deep Convolutional Neural Networks for IoT applications is proposed. CNN is used to extract the deep iris features from both left and right irises which will feed a fully connected neural network with a Softmax classifier. Three publicly available datasets, namely CASIA V4-Interval, Phoenix, and UBIRIS V1, are used during experiments for training and testing the proposed model.

The proposed system shows satisfied and competitive results regard reliability, the accuracy of recognition rate, training time, and robustness among a lot of state-of-the-art methods. The accuracy of the recognition rate of the proposed model is 99.24%,100%, and 98% with CASIA V4, Phoenix, and UBIRIS V1 datasets, respectively. The training time of the Phoenix dataset was about 17 minutes for each right and left sub-sets, 22 minutes for UBIRIS. V1 dataset and the training time of the CASIA V4 interval dataset was about 25 minutes for each right and left iris sub-sets.

The results of the proposed system are highly recommended to be a solution for the security issues of many IoT systems, especially in the era in which IoT applications are considered the source of the biggest kind of information on the internet.

A chaotic encryption algorithm based on a key sequence, generated by a sequence of logistic maps and sequences of states of (LFSR), is used to secure iris template transmission over the internet. We also showed how well the proposed system deals with noised iris images from Gaussian (Normal) and the uniform distributions. The proposed system shows a low degradation of recognition accuracy rates in the case of using noise from Gaussian and the uniform distributions with different standard deviations and intervals, respectively. Also, an efficient iris recognition system for one of the most important real-life IoT applications, which is ATMs for banking systems based on the mobile application, is proposed. A bank mobile application is implemented to generate OTP for ATMs which increases the overall system defense.

6.2 Future Work

In the future, our plans include the following:

- Applying the proposed system in other real-life IoT applications such as airport travelers' recognition systems.
- Also, the implementation of iris recognition systems for mobile phones using databases created using smartphone devices like VISOB [94], [95],CASIA Iris M1 (mobile) [96]–[98], CASIA BTAS [99], MICHE DB [100]–[102]and CSIP [103], [104].
- We also may build recognition systems for other biometrics especially behavioral biometrics like gait and lip motion.

References

[1] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide to Biometrics*, 1st ed. Springer-Verlag New York, 2004.

[2] K. W. Bowyer and M. J. Burge, Eds., *Handbook of Iris Recognition*, 1st ed. Springer-Verlag London, 2013.

[3] A. K. Jain, A. A. Ross, and K. Nandakumar, *Introduction to Biometrics*, 1st ed. Springer US, 2011.

[4] R. Singhal, N. Singh, and P. Jain, “Towards an Integrated Biometric Technique,” *Int. J. Comput. Appl.*, vol. 42, no. 13, pp. 20–23, 2012.

[5] D. Maltoni, R. Cappelli, and D. Meuwly, “Handbook of Biometrics for Forensic Science,” *Springer*, 2017.

[6] A. Ross and A. K. Jain, “Multimodal biometrics: An overview,” in *European Signal Processing Conference*, 2004, pp. 1221–1224.

[7] S. P. A. K. Jain, r. Bolle, *Biometrics: Personal Identification in Networked Security*. 1999.

[8] C. Rathgeb, Andreas Uhl, and P. Wild, *Iris recognition: from segmentation to template security*, 1st ed. Springer, 2012.

[9] A. K. Ross, Arun A., Nandakumar, Karthik, Jain, *Handbook of Multibiometrics*, 1st ed. Springer, 2006.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning*, 2nd ed. Springer-Verlag New York, 2009.

[11] G. James, D. Witten, T. Hastie, and R. Tibishirani, *An Introduction to Statistical Learning with Applications in R*, 1st ed. Springer-Verlag New York, 2013.

[12] T. M. Mitchell, *Machine Learning*, 1st ed. McGraw-Hill Education, 1997.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer-Verlag New York, 2006.

[14] M. I. Jordan and C. M. Bishop, “Neural networks,” in *Computer Science Handbook, Second Edition*, 2004.

[15] Simon Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson, 2008.

[16] S. Minaee, A. Abdolrashidiy, and Y. Wang, “An experimental study of deep convolutional features for iris recognition,” in *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2016, pp. 37–42.

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

[18] “VGG16 architecture.” https://neurohive.io/en/popular-networks/vgg16/ (accessed Oct. 17, 2020).

[19] “ResNet architecture.” https://neurohive.io/en/popular-networks/resnet/ (accessed Oct. 17, 2020).

[20] “Inception V3 architecture.” https://keras.io/api/applications/inceptionv3/ (accessed Oct. 17, 2020).

[21] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep Learning Approach for Intelligent Intrusion Detection System,” *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[22] C. C. Aggarwal, *Neural Networks and Deep Learning - A Textbook*, 1st ed. Springer International Publishing, 2018.

[23] F. Gaxiola, P. Melin, and M. Lopez, “Modular neural networks for person recognition using segmentation and the iris biometric measurement with image pre-processing,” in *Proceedings of the International Joint Conference on Neural Networks*, 2010.

[24] A. S. Al-Waisy, R. Qahwaji, S. Ipson, S. Al-Fahdawi, and T. A. M. Nagem, “A multi-biometric iris recognition system based on a deep learning approach,” *Pattern Anal. Appl.*, vol. 21, pp. 783–802, 2018.

[25] “CASIA iris dataset.” http://www.cbsr.ia.ac.cn/china/Iris Databases CH.asp (accessed Jan. 23, 2021).

[26] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 7th ed. Pearson Education, Inc., 2016.

[27] W. Stallings, *Computer Security: Principles and Practice*, 4th ed. Pearson, 2017.

[28] G. Mehta, M. K. Dutta, and P. S. Kim, “A secure encryption method for biometric templates based on chaotic theory,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2016.

[29] F. Dachselt and W. Schwarz, “Chaos and cryptography,” *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 48, no. 2, pp. 1498–1508, 2001.

[30] M. Dobeš, J. Martinek, D. Skoupil, Z. Dobešová, and J. Pospíšil, “Human eye localization using the modified Hough transform,” *Optik (Stuttg).*, vol. 117, no. 10, pp. 468–473, 2006.

[31] M. Dobeš, L. Machala, P. Tichavský, and J. Pospíšil, “Human eye iris recognition using the mutual information,” *Optik (Stuttg).*, vol. 115, no. 9, pp. 399–404, 2004.

[32] “Phoenix iris Dataset.” http://phoenix.inf.upol.cz/iris/ (accessed Nov. 06, 2019).

[33] H. Proença and L. A. Alexandre, “UBIRIS: A noisy iris image database,” in *International Conference on Image Analysis and Processing*, 2005, pp. 970–977.

[34] “UBIRIS. V1 iris Dataset.” http://iris.di.ubi.pt/index_arquivos/Page374.html (accessed Sep. 14, 2020).

[35] S. Rohith, K. N. H. Bhat, and A. N. Sharma, “Image encryption and decryption using

chaotic key sequence generated by sequence of logistic map and sequence of states of Linear Feedback Shift Register," in *International Conference on Advances in Electronics, Computers and Communications (ICAECC) 10-11 Oct.*, 2014.

[36] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, 9th ed. Pearson Education, Inc., 2012.

[37] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 6th ed. John Wiley & Sons, Inc., 2014.

[38] A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-Fahdawi, "A robust face recognition system based on Curvelet and Fractal dimension transforms," in *Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se*, 2015, pp. 548–555.

[39] R. Clarke, "Human Identification in Information Systems: Management Challenges and Public Policy Issues," *Inf. Technol. People*, vol. 7, no. 4, pp. 6–37, 1994.

[40] Z. Sun and T. Tan, "Ordinal measures for iris recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2211–2226, 2009.

[41] R. Hentati, M. Hentati, and M. Abid, "Development a New Algorithm for Iris Biometric Recognition," *Int. J. Comput. Commun. Eng.*, vol. 1, no. 3, pp. 283–2, 2012.

[42] R. Hidayat and K. Ihsan, "Robust Feature Extraction and Iris Recognition for Biometric Personal Identification," in *Biometric Systems, Design and Applications*, 2011.

[43] O. Vermesan *et al.*, "Internet of Things Strategic Research Roadmap," *Internet Things Glob. Technol. Soc. Trends*, vol. 1, pp. 9–25, 2011.

[44] I. Pena-Lopez, "Itu Internet Report 2005: The Internet of Things," 2005.

[45] M. Zamfir, V. Florian, A. Stanciu, G. Neagu, Ş. Preda, and G. Militaru, "Towards a platform for prototyping IoT health monitoring services," in *Lecture Notes in Business Information Processing*, 2016.

[46] N. E. Oweis, C. Aracenay, W. George, M. Oweis, H. Soori, and V. Snasel, "Internet of things: Overview, sources, applications and challenges," in *Advances in Intelligent Systems and Computing*, 2016.

[47] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*. 2017.

[48] I. Mashal, O. Alsaryrah, T. Y. Chung, C. Z. Yang, W. H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015.

[49] O. Said and M. Masud, "Towards internet of things: Survey and future vision," *Int. J. Comput. Networks*, vol. 5, no. 1, pp. 1–17, 2013.

[50] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, and H. Y. Du, "Research on the architecture of

Internet of Things," in *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, 2010, pp. 484–487.

[51] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," in *Proceedings - 10th International Conference on Frontiers of Information Technology, FIT 2012*, 2012, pp. 257–260.

[52] S. Lee, H. Kim, D. K. Hong, and H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," in *International Conference on Information Networking*, 2013.

[53] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[54] I. Arel, D. Rose, and T. Karnowski, "Deep machine learning-A new frontier in artificial intelligence research," *IEEE Comput. Intell. Mag.*, 2010.

[55] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Face recognition using convolutional neural network and simple logistic classifier," in *Advances in Intelligent Systems and Computing*, 2014, pp. 197–107.

[56] A. Courville, I. Goodfellow, and Y. Bengio, *Deep Learning*, 1st ed. The MIT Press, 2016.

[57] J. G. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, 1993.

[58] X. Ren *et al.*, "An improved method for Daugman's iris localization algorithm," *Comput. Biol. Med.*, vol. 38, no. 1, pp. 111–115, 2008.

[59] S. A. Sahmoud and I. S. Abuhaiba, "Efficient iris segmentation method in unconstrained environments," *Pattern Recognit.*, vol. 46, no. 12, pp. 3174–3185, 2013.

[60] H. Proença and L. A. Alexandre, "Iris segmentation methodology for non-cooperative recognition," *IEE Proc. Vision, Image Signal Process.*, vol. 153, no. 2, pp. 199–205, 2006.

[61] R. P. Wildes, "Iris recognition: An emerging biometrie technology," *Proc. IEEE*, vol. 85, no. 9, pp. 1348–1363, 1997.

[62] W. W. Boles and B. Boashash, "A Human Identification Technique Using Images of the Iris and Wavelet Transform," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1185–1188, 1998.

[63] S. Lim, K. Lee, O. Byeon, and T. Kim, "Efficient iris recognition through improvement of feature vector and classifier," *ETRI J.*, vol. 23, no. 2, pp. 61–70, 2001.

[64] L. Masek and P. Kovesi, "Recognition of human iris patterns for biometric identification," *The School of Computer Science and Software*. 2003.

[65] M. Zeng *et al.*, "Convolutional Neural Networks for human activity recognition using mobile sensors," in *Proceedings of the 2014 6th International Conference on Mobile Computing, Applications and Services, MobiCASE 2014*, 2015, pp. 197–205.

[66] A. R. Syafeeza, M. Khalil-Hani, S. S. Liew, and R. Bakhteri, “Convolutional neural network for face recognition with pose and illumination variation,” *Int. J. Eng. Technol.*, vol. 6, no. 1, pp. 44–57, 2014.

[67] J. Lozej, D. Stepec, V. Struc, and P. Peer, “Influence of segmentation on deep iris recognition performance,” in *7th International Workshop on Biometrics and Forensics, IWBF*, 2019.

[68] “AlexNet architecture.” https://neurohive.io/en/popular-networks/alexnet-imagenet-classification-with-deep-convolutional-neural-networks/ (accessed Oct. 17, 2020).

[69] S. S. Dhage, S. S. Hegde, K. Manikantan, and S. Ramachandran, “Dwt-based feature extraction and Radon transform based contrast enhancement for improved iris recognition,” *Procedia Comput. Sci.*, vol. 45, pp. 256–265, 2015.

[70] H. K. Rana, M. S. Azam, M. R. Akhtar, J. M. W. Quinn, and M. A. Moni, “A fast iris recognition system through optimum feature extraction,” *PeerJ Comput. Sci. 8 April*, 2019.

[71] R. M. Sundaram and B. C. Dhara, “Neural network based iris recognition system using Haralick features,” in *3rd International Conference on Electronics Computer Technology (ICECT)*, 2011, pp. 19–23.

[72] B. V. Bharath, A. S. Vilas, K. Manikantan, and S. Ramachandran, “Iris recognition using radon transform thresholding based feature extraction with Gradient-based Isolation as a pre-processing technique,” in *9th International Conference on Industrial and Information Systems (ICIIS),* Dec. 2014, pp. 1–8.

[73] R. Khanam, Z. Haseen, N. Rahman, and J. Singh, “Performance analysis of iris recognition system,” *Adv. Intell. Syst. Comput.*, vol. 847, pp. 159–171, 2019.

[74] G. Singh, R. K. Singh, R. Saha, and N. Agarwal, “IWT Based Iris Recognition for Image Authentication,” *Procedia Comput. Sci.*, vol. 171, pp. 1868–1876, 2020.

[75] K. Saminathan, T. Chakravarthy, and M. Chithra Devi, “Iris recognition based on kernels of Suppoet Vector Machine,” *ICTACT J. Soft Comput.*, vol. 5, no. 2, pp. 889–895, 2015.

[76] R. Gad, A. A. Abd El-Latif, S. Elseuofi, H. M. Ibrahim, M. Elmezain, and W. Said, “IoT Security Based on Iris Verification Using Multi-Algorithm Feature Level Fusion Scheme,” in *2nd International Conference on Computer Applications and Information Security, ICCAIS 2019*, 2019.

[77] R. Gad, M. Talha, A. EL-SAYED, N. EL-Fishawy, G. Muhammad, and M. Zorkany, “Iris Recognition Using Multi-Algorithmic Approaches for Cognitive Internet of things (CIoT) Framework,” *Futur. Gener. Comput. Syst.*, vol. 89, pp. 178–191, 2018.

[78] R. Gad, M. Zorkany, A. EL-Sayed, and N. EL-Fishawy, “An Efficient Approach for Simple Iris Localization and Normalization Technique,” *Menoufia J. Electron. Eng. Res.*, vol. 25, no. 2, pp. 213–224, 2016.

[79] H. Habibi Aghdam and E. Jahani Heravi, *Guide to Convolutional Neural Networks,* 1st ed. Springer International Publishing, 2017.

[80] S. Sriram, R. Vinayakumar, V. Sowmya, M. Alazab, and K. P. Soman, "Multi-scale learning based malware variant detection using spatial pyramid pooling network," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2020*, 2020, pp. 740–745.

[81] "Raspberry Pi 2." https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ (accessed Sep. 14, 2020).

[82] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration," *ACM Comput. Surv.*, vol. 51, no. 6, 2019.

[83] W. R. Stevens, B. Fenner, A. M. Rudoff, and K. Juszkiewicz, *Unix Network Programming Volume 1: The Sockets Networking API*, 3rd ed. Addison-Wesley Professional, 2003.

[84] T. S. Parker and L. Chua, *Practical Numerical Algorithms for Chaotic Systems*, 1st ed. Springer-Verlag New York, 1989.

[85] M. M. R. Mano and M. D. Ciletti, *Digital Design*, 5th ed. Pearson, 2013.

[86] R. A. Brualdi, *Introductory Combinatorics.*, 5th ed. Pearson Education, Inc., 2009.

[87] V. K. ROHATGI and A. K. M. E. SALEH, *AN INTRODUCTION TO PROBABILITY AND STATISTICS*, 3rd ed. Canada: John Wiley & Sons, Inc., 2015.

[88] J. A. Rice, *Mathematical Statistics and Data Analysis*, 3rd ed. Thomson Learning, Inc., 2007.

[89] Sheldon Ross, *A first course in probability*, 9th ed. Pearson Education, Inc., 2014.

[90] S. Kurnia, F. Peng, and Y. R. Liu, "Understanding the adoption of electronic banking in China," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2010, pp. 1–10.

[91] M. Vrîncianu and L. A. Popa, "Considerations Regarding the Security and Protection of E-Banking Services Consumers' Interests," *AMFITEATRU Econ. J.*, vol. 12, no. 28, pp. 388–403, 2010.

[92] L. Peotta, M. D. Holtz, B. M. David, F. G. Deus, and R. Timoteo de Sousa, "A Formal Classification of Internet Banking Attacks and Vulnerabilities," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 186–197, 2011.

[93] B. Chaimaa, E. Najib, and H. Rachid, "E-banking Overview: Concepts, Challenges and Solutions," *Wirel. Pers. Commun.*, pp. 1059–1078, 2020.

[94] "Visible light mobile Ocular Biometric (VISOB) Dataset ICIP2016 Challenge Version." Visible light mobile Ocular Biometric (VISOB) Dataset ICIP2016 Challenge Version (accessed Feb. 27, 2021).

[95] A. Rattani, R. Derakhshani, S. K. Saripalle, and V. Gottemukkula, "ICIP 2016 competition on mobile ocular biometric recognition," in *Proceedings - International Conference on Image Processing, ICIP*, 2016, pp. 320–324.

[96] “CASIA-Iris-Mobile-V1.0 - Casia mobile database (datasets S1, S2 and S3).” http://biometrics.idealtest.org/dbDetailForUser.do?id=13 (accessed Feb. 27, 2021).

[97] Q. Zhang, H. Li, Z. Sun, and T. Tan, “Deep feature fusion for iris and periocular biometrics on mobile devices,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2897–2912, 2018.

[98] Q. Zhang, H. Li, M. Zhang, Z. He, Z. Sun, and T. Tan, “Fusion of face and iris biometrics on mobile devices using near-infrared images,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, pp. 569–578.

[99] Z. Man, Q. Zhang, Z. Sun, S. Zhou, and N. U. Ahmed, “The BTAS Competition on Mobile Iris Recognition,” in *8th International Conference on Biometrics Theory, Applications and Systems (BTAS 2016)*, 2016, pp. 1–7.

[100] M. De Marsico, M. Nappi, F. Narducci, and H. Proença, “Insights into the results of MICHE I - Mobile Iris CHallenge Evaluation,” *Pattern Recognit.*, vol. 74, pp. 286–304, 2018.

[101] M. De Marsico, M. Nappi, D. Riccio, and H. Wechsler, “Mobile Iris Challenge Evaluation (MICHE)-I, biometric iris dataset and protocols,” *Pattern Recognit. Lett.*, vol. 57, pp. 17–32, 2015.

[102] “MICHE - Mobile Iris CHallenge Evaluation.” http://biplab.unisa.it/MICHE/ (accessed Feb. 27, 2021).

[103] “Cross Sensor Iris and Periocular Database.” http://csip.di.ubi.pt/ (accessed Feb. 27, 2021).

[104] G. Santos, E. Grancho, M. V. Bernardo, and P. T. Fiadeiro, “Fusing iris and periocular information for cross-sensor recognition,” *Pattern Recognit. Lett.*, vol. 57, pp. 52–59, 2015.

ولأن ماكينات الصراف الآلي (ATMs) تعد من أهم بيئات إنترنت الأشياء. لأن التحقق من هوية عملاء البنوك من خلال ماكينات الصراف الآلي يعد من المهمات بالغة الأهمية حيث تحتاج أموال العملاء وبياناتهم ومعلوماتهم المهمة إلى حماية عالية. طرق التحقق الحالية الموجودة في أجهزة الصراف الآلي التي تتم باستخدام مزيج من بطاقة الائتمان وكلمة المرور لها الكثير من العيوب مثل السطو والسرقة وانتهاء الصلاحية وحتى الضياع المفاجئ. ولأن دمج نظام فعال للتعرف على قزحية العين في بيئات إنترنت الأشياء الحرجة مثل أجهزة الصراف الآلي قد يتضمن العديد من السيناريوهات المعقدة. لمعالجة هذه المشكلات، في هذه الرسالة أيضا تم بناء نظام مصادقة كامل وفعال لأجهزة الصراف الآلي بشكل عملي يعتمد على تطبيق الهاتف المحمول الخاص بالبنك ونظام التعرف على قزحية العين القائم على مجوعة بيانات (datasets) مختلفة من حيث أساليب الالتقاط مثل (Visible light vision datasets) و(Near-Infrared datasets).

<u>وقد تضمنت الرسالة ستة أبواب تم تنظيمها على النحو التالى:</u>

- يقدم الباب الأول مقدمة عامة عن دوافع هذه الرسالة والمشكلات الرئيسية التي تحاول حلها والاسهامات الرئيسية لحل هذه المشاكل.
- يقدم الفصل الثاني مراجعة عامة حول القياسات الحيوية، والتعرف على قزحية العين القائم على إنترنت الأشياء، وأساسيات الشبكات العصبية والتعلم العميق.
- يركز الفصل الثالث على المكونات الأساسية لنظام التعرف على قزحية العين بشكل عام دون التعرض الي تطبيقه في بيئات إنترنت الأشياء، وفي هذا الفصل أيضا يتم توصيف المنهجيات المستخدمة لتجزئة وتطبيع قزحية العين واستخراج الميزات منها والتصنيف، ويناقش أخيرًا النتائج ذات الصلة بكل هذه المنهجيات.
- يناقش الفصل الرابع كيفية تطوير النظام المقترح ليناسب بيئة إنترنت الأشياء العامة، ويصف منهجية الاتصال المستخدمة، ويناقش أيضا تشفير قزحية العين وفك تشفيرها عبر قنوات الاتصال، ودراسة تأثير إضافة أنواع مختلفة من الضوضاء إلى صور قزحية العين لتقييم موثوقية النظام المقترح.
- يقدم الفصل الخامس سيناريو واقعي من بيئات إنترنت الأشياء وهو عبارة عن أجهزة الصراف الآلي للأنظمة المصرفية ويناقش كيفية تضمين نظام التعرف على قزحية العين المقترح فيها.
- يلخص الفصل السادس نتائج النظام المقترح وخطط الأعمال المستقبلية.

وقد ذيلت الرسالة بالمراجع العلمية التي تم الاستعانة بها.

التعرف على قزحية العين لأمن انترنت الأشياء

ملخص الرسالة

في الآونة الأخيرة، تلعب أنظمة التأمين القائمة على القياسات الحيوية دورًا حيويًا في نجاح تأمين تطبيقات إنترنت الأشياء(IoT) . تحل قزحية العين الكثير من مشكلات الأمان، لا سيما في التطبيقات الذكية القائمة على إنترنت الأشياء. فهي تقوم بزيادة مقاومة هذه الأنظمة ضد الهجمات. في هذه الرسالة، تم اقتراح نظام فعال للتعرف على قزحية العين يعتمد على الشبكات العصبية التلافيفية العميقة (CNN) من أجل التطبيقات المختلفة لإنترنت الأشياء. يتم استخدام الشبكات العصبية التلافيفية لاستخراج خصائص القزحية من كلتا العينين اليمنى واليسرى مما يزيد من الوثوق بالنظام المقترح في التعرف على الأشخاص وهذا ما تجاهله معظم الباحثين العاملين في مجال التعرف على قزحية العين، حيث قاموا ببناء نماذج التصنيف الخاصة بهم بناءً على قزحية واحدة فقط إما اليسرى أو اليمني مما يؤدي إلى تقليل الوثوق في تلك النظم ضد الهجمات في أنظمة الحياة الواقعية. بعد ذلك يتم استخدام خصائص القزحية كوحدات إدخال للشبكة العصبية المتصلة بالكامل مع مصنف (Softmax). تُستخدم مجموعة بيانات (Phoenix dataset) و (CASIA V4 Interval dataset) و (UBIRIS V1 dataset) لتدريب نظام الشبكات العصبية التلافيفية للحصول على أفضل ضبط لمعلمات الشبكة. وقد أظهرت النتائج أن النظام المقترح للتعرف على قزحية العين يحقق دقة كبيرة مقارنة بالطرق الحالية، حيث تم الحصول على 98% و99.24% و100% مع (UBIRIS V1 dataset) و (CASIA V4 Interval dataset) و(Phoenix dataset) على الترتيب. وبالتالي النظام المقترح بالرسالة يحقق نتائج مرضية وتنافسية فيما يتعلق بالدقة والاستقرار مقارنة بالأساليب الحالية. كما تتميز الطريقة المقترحة بوقت تدريب للبيانات منخفض نسبيًا، وهو عامل مفيد في التطبيقات الهامة القائمة على إنترنت الأشياء.

في هذه الرسالة أيضا تم دمج النظام المقترح للتعرف على قزحية العين بشكل عملي في نظام أكبر لبيئة عامة مقترحة لإنترنت الأشياء تم بنائها باستخدام بوردة الأنظمة المدمجة (raspberry pi 2). وقد تم أيضا مناقشة تأثير إضافة أنواع مختلفة من الضوضاء إلى صور قزحية العين التي قد تحدث نتيجة الضوضاء المرتبطة بأجهزة الاستشعار الخاصة بتطبيقات إنترنت الأشياء أو الالتقاط السيئ لصور قزحية العين من قبل مستخدمي النظام أو أي نوع أخر من الاعتداءات التي من الممكن ان يتعرض لها النظام، حيث تم مناقشة تأثير نوعين مختلفين من الضوضاء، النوع الأول مولد عشوائيا من توزيع (Gaussian) والثاني من توزيع (Uniform). وقد تم استخدام التشفير العشوائي لتأمين نقل الصور الخاصة بقزحية العين في النظام المقترح. وبالنسبة لما يتعلق بمعدل دقة التعرف على القزحية في حالة دمج نظام التعرف السابق ذكره في البيئة المقترحة لإنترنت الأشياء، تُظهر هذه الطريقة انخفاضا طفيفا لمعدلات دقة التعرف في حالة استخدام صور قزحية مشوشة بأنواع الضوضاء السابق ذكرها.

جامعة المنوفية
كلية الهندسة الالكترونية بمنوف
قسم هندسة وعلوم الحاسبات

التعرف على قزحية العين لأمن انترنت الأشياء

رسالة مقدمة للحصول على درجة الماجيستير في العلوم الهندسية
تخصص هندسة وعلوم الحاسبات
مجال الرسالة: الذكاء الاصطناعي ومعالجة الصور
قسم هندسة وعلوم الحاسبات

مقدمة من

أحمد صبرى عبد الخالق شلبى

معيد بقسم هندسة وعلوم الحاسبات – كلية الهندسة الالكترونية بمنوف
بكالوريوس الهندسة الالكترونية – تخصص هندسة وعلوم الحاسبات
كلية الهندسة الالكترونية بمنوف

لجنة الحكم والمناقشة

أ.د./ نوال أحمد الفيشاوي (   )
أستاذ متفرغ بقسم هندسة وعلوم الحاسبات
كلية الهندسة الإلكترونية بمنوف

أ.د./ أمانى محمود سرحان (   )
أستاذ ورئيس قسم هندسة الحاسبات والتحكم الآلي
كلية الهندسة ـ جامعة طنطا

أ.م.د./ نرمين عبدالوهاب البهنساوي (   )
أستاذ مساعد بقسم هندسة وعلوم الحاسبات
كلية الهندسة الإلكترونية بمنوف ـ جامعة المنوفية

وكيل الكلية للدراسات العليا والبحوث
أ.د./ مني محمد صبري شقير

2021

جامعة المنوفية
كلية الهندسة الالكترونية بمنوف
قسم هندسة وعلوم الحاسبات

التعرف على قزحية العين لأمن انترنت الأشياء

رسالة مقدمة للحصول على درجة الماجيستير في العلوم الهندسية
تخصص هندسة وعلوم الحاسبات
مجال الرسالة: الذكاء الاصطناعي ومعالجة الصور
قسم هندسة وعلوم الحاسبات

مقدمة من

أحمد صبرى عبد الخالق شلبى

معيد بقسم هندسة وعلوم الحاسبات – كلية الهندسة الالكترونية بمنوف
بكالوريوس الهندسة الالكترونية – تخصص هندسة وعلوم الحاسبات
كلية الهندسة الالكترونية بمنوف

لجنة الإشراف

أ.د./ نوال أحمد الفيشاوي (   )
أستاذ متفرغ بقسم هندسة وعلوم الحاسبات
كلية الهندسة الإلكترونية بمنوف

د./ عز الدين بدوي جاد الرب حمدان (   )
مدرس بقسم هندسة وعلوم الحاسبات
كلية الهندسة الإلكترونية بمنوف

2021

جامعة المنوفية
كلية الهندسة الالكترونية بمنوف
قسم هندسة وعلوم الحاسبات

التعرف على قزحية العين لأمن انترنت الأشياء

رسالة مقدمة للحصول على درجة الماجيستير في العلوم الهندسية
تخصص هندسة وعلوم الحاسبات
مجال الرسالة: الذكاء الاصطناعي ومعالجة الصور
قسم هندسة وعلوم الحاسبات

مقدمة من

أحمد صبرى عبد الخالق شلبى

معيد بقسم هندسة وعلوم الحاسبات – كلية الهندسة الالكترونية بمنوف
بكالوريوس الهندسة الالكترونية – تخصص هندسة وعلوم الحاسبات
كلية الهندسة الالكترونية بمنوف

لجنة الإشراف

أ.د./ نوال أحمد الفيشاوي

أستاذ متفرغ بقسم هندسة وعلوم الحاسبات
كلية الهندسة الإلكترونية بمنوف

د./ عز الدين بدوي جاد الرب حمدان

مدرس بقسم هندسة وعلوم الحاسبات
كلية الهندسة الإلكترونية بمنوف

2021