Republic of Iraq
Ministry of Higher Education and Science Research
University of Babylon
Collage of Science for Women
Department of Computer Science

Human Multi Biometric Fusion using Deep Learning Techniques

A Thesis

Submitted to the Council of College of Science for Women, University of Babylon in Partial Fulfillment of the Requirement for Degree of Master in Computer Science

By

Hawraa Abed Al-kareem Hussain

Supervised by

Prof. Dr. Hawraa Hassan Abbas

2023 A.D. **1445 A.H**

بسم الله الرحمن الرحيم

قَالُواْ سُبْحَانَكَ لاَ عِلْمَ لَنَا إِلاَّ مَا عَلَّمْتَنَا

إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ

صَدَقَ اللهُ العَلِيُّ العَظيم

سورة البقرة (٣٢)

CERTIFICATION OF THE EXAMINATION COMMITTEE

We are the members of the examination committee, certify that we have read this thesis entitled (**Human Multi Biometric Fusion using Deep Learning Techniques**) and after the examining the master student **(Hawraa Abed Al-kareem Hussain)** in its contents in 1/8/2023 and that in our opinion it is adequate as a thesis for the degree of Master in Science \ Computer Science with degree (Very Good).

Committee Chairman

Signature:

Name: **Samaher Hussein Ali Al-Janabi**

Scientific order: Prof. Dr.

Address: University of Babylon / College of Science for Women

Date: / / 2023

Committee Member (**Supervisor**)

Signature:

Name: **Hawraa Hassan Abbas**

Scientific order: Prof. Dr.

Address: University of Kerbala /College of Engineering

Date: / / 2023

Committee Member

Signature:

Name: **Ayad Rodhan Abbas**

Scientific order: Prof. Dr.

Address: University of Technology / Department of Computer Science

Date: / / 2023

Committee Member

Signature:

Name: **Salah Mahdi Saleh**

Scientific order: Lecturer. Dr.

Address: University of Babylon /College of Science for Women

Date: / / 2023

Date of Examination: 1/ 8 /2023

Deanship Authentication of college of Science for Women

Approved for the College Committee of graduate studies.

Signature

Name: **Abeer Fauzi Al-Rubaye**

Scientific Order: **Prof. Dr.**

Address: **Dean of College of Science for Women**

Date: / /2023

Supervisor Certification

I certify that project entitled **"Human Multi Biometric Fusion using Deep Learning Techniques"** was prepared at the Department of computer Sciences/ College of Science for Women/ University of Babylon, by (Hawraa Abed Al-kareem Hussain) as partial fulfilment of the requirements for the degree of Master in Computer Science.

Signature:
Name: Prof. Dr. Hawraa Hassan abbas
(Supervisor)

Date: / /2023

Address: Department of Electrical & Electronic Engineering, College of Engineering, University of Kerbala, Iraq

The Head of the Department Certification

In view of the available recommendations, I forward the research entitled "**Human Multi Biometric Fusion using Deep Learning Techniques**" for debate by the examination committee.

Signature:
Name: Assist. Prof. Dr. Saif Al-alak
Date: / / 2023
Address: University of Babylon/College of Science for Women

Dedication

To the spring of tenderness, my dear Parents and all my family

Hawraa Abed Al-kareem Hussain

2023

Acknowledgments

First of all, great thanks to Allah whose will and direction made it possible for me to finish my thesis.

I would like to express my sincere gratitude to my supervisor Prof. Dr. Hawraa Hassan Abbas for her advice, support, continuous encouragement, direction's opinion and supervision throughout this work to be in the best manner.

Additionally, I want to thank her for giving me the chance to work in the disciplines of deep learning and artificial intelligence.

In addition, many thanks and love to all of my family members who have supported me in completing my research and have shown me nothing but love and sincere affection.

Finally, I would like to thank the people who helped me.

Hawraa Abed Al-kareem Hussain

2023

Abstract

Since spoofing attacks have increased significantly, a lot of research is being done on biometric security systems. Researchers are more interested in multimodal biometrics to give better security using biometric applications.

Human biometric fusion is an approach that combines multiple biometric modalities. A deep learning model is designed to identify people using their face, iris, and fingerprint biometrics.

The Convolutional Neural Networks (CNNs) that form the model's structure extract features from the images and use the Softmax classifier to categorize them.

Many fusion approaches were tested to fuse the CNN models to examine their influence on recognition performance. As a result, score and feature level fusion approaches were used. In addition, CNN algorithms (VGG16, ReseNet50, MobilevNet, DenseNet, GoogleNet) are implemented to create the additional single modal face, iris, and fingerprint biometric models to compare their results and prove the efficiency and superiority of the multimodal model.

. Model training and testing as well as model evaluation use the publicly accessible actual multimodal biometric dataset SDUMLA-HMT. With a feature-level fusion technique, the proposed model accuracy was 97.55%, and with a score-level fusion approach, it was 99.37%, easily outperforming existing state-of-the-art methods, according to the results.

Thus, the proposed multi-biometric identification model could be representing a robust authorization technique

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OFTABLES

LIST OF ABBREVIATIONS

Abbreviation	*Meaning*
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under Curve
BF	Biometrics Fusion
BP	Backpropagation
CNN	Convolutional Neural Network
DA	Data Augmentation
DL	Deep Learning
DNN	Deep Neural Network
FC	Fully Connected
FFNs	Feedforward Neural Networks
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
ML	Machine Learning
ReLU	Rectified Linear Unit
R-HOG	Rectangle Histogram of Oriented Gradient
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group

Chapter One
General Introduction

1.1 Introduction

The science and technology of biometrics is the measurement and analysis of biological data from the human body for enhancing system security by providing precise and dependable patterns and algorithms for person identification and verification. Governments, the military, and businesses all use biometrics solutions extensively. For person identification or recognition system, various types of biometrics traits can be used.

All biometrics traits are grouped into physiological, behavioral, and soft biometrics [1]. Under these three categories, various biometric traits are classified. Fig. 1.1 shows three groups of biometric traits. From these categories of biometrics, more than one biometric trait can be combined in the multimodal biometric recognition systems.

Unimodal systems, which are a unique source of data in biometric systems, are flawless but frequently have several issues when dealing with noisy data, such as intra-class variances, constrained degrees of freedom, and non-universality. Utilizing multimodal biometric systems, also known as "Biometric Fusion" systems, which combine two or more biometrics, can help alleviate a number of these issues. Due to the high degree of sophistication in the formation's architecture, a fused biometric classification is superior to a unimodal biometric classification. Information fusion in multimodal systems can be accomplished using a variety of techniques, fusion levels, and integration methodologies.

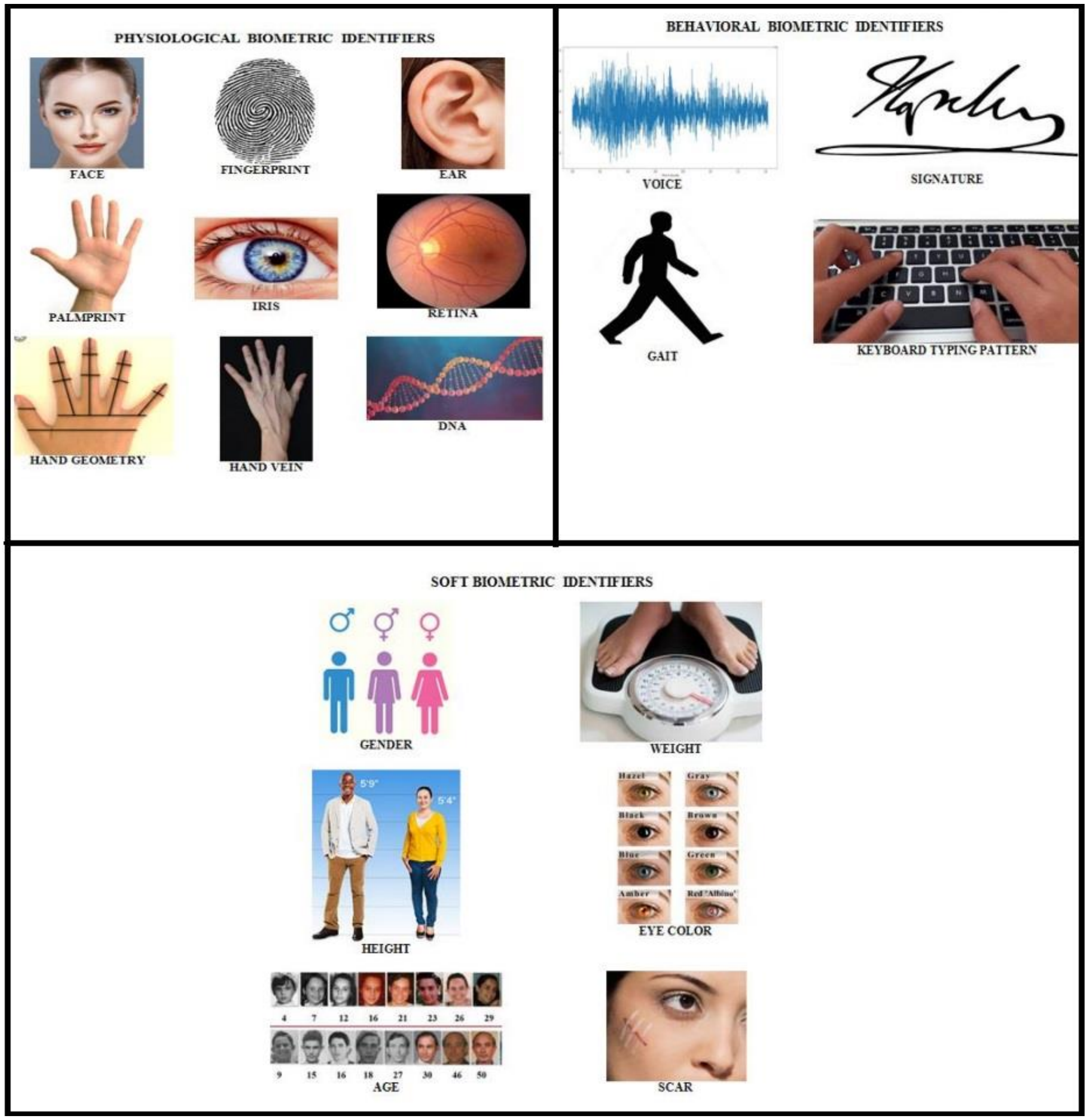


Figure 1.1 Biometric Traits [2]

A biometric recognition system typically consists of four stages: sensors, extraction of features, matching, and identification decision making [3]. In essence, multimodal biometric systems need more than one characteristic to identify the person [3]. Information that can be accessed in a biometric system module can be used to combine many quantities in multimodal biometric systems.

For purposes of recognition, several biometrics researchers have used machine learning techniques [4]. Before classifying the raw biometric data, machine learning algorithms need to be transform it into a suitable format (do some prepossessing steps such as scaling and rotating) and then extract characteristics or features from it [5].

Recently deep learning has a significant impact and delivered outstanding outcomes in biometrics systems. Many of the drawbacks of traditional machine learning algorithms, especially those related to feature extraction methods, have been solved by deep learning algorithms. Deep learning algorithms can handle the transformation of biometric images and features from raw data [6].

This research intends to thoroughly evaluate the algorithm for convolutional neural networks (CNN) performance in recognizing a person using three biometric attributes, namely face, iris, and fingerprint. These characteristics were chosen because the fingerprint is probably the most well-known and evident individual recognition attribute, and the iris is a good alternative due to its distinctive and extremely exact nature of recognition information. The third trait, face, has been incorporated to increase the accuracy of identification outcomes as well as the security and dependability of the suggested model.

1.2 Problem Statement

1.2.1 Programming challenges

Each biometric modality may have different data formats, resolutions, and quality. Preprocessing the data to bring it to a consistent format and quality level can be a challenging task. Different biometric modalities require specific algorithms and techniques for feature extraction. Each modality may have unique characteristics and require domain-specific knowledge. Developing efficient feature extraction methods for each modality and representing them in a unified format for fusion is a complex programming task.

1.2.2 Application Challenges

Integrating multibiometric fusion systems into existing applications or infrastructure can be challenging. The systems need to be compatible with the existing software and hardware components, which may have different data formats, protocols, and interfaces. Ensuring seamless integration and interoperability requires careful planning and programming.

1.3 Hypothesis of thesis

In order to verify or identify individuals for authorization problems, multi-biometric systems use the evidence provided by various biometric sources (such as the face, iris, and fingerprint). The purpose of this thesis is to create reliable multimodal biometric model for identity verification that fuses the modalities of face, iris, and fingerprint at the feature and score levels. . It is also more user-friendly, producing accurate results, attack-resistant, and matching data quickly.

1.4 Limitation of thesis

1. Increased complexity: Multibiometric fusion systems are inherently more complex than single-modality systems. Integrating multiple biometric modalities, preprocessing the data, developing fusion algorithms, and managing the system architecture require additional complexity in design, implementation, and maintenance. This complexity can make system development and management more challenging.

2. Hardware and infrastructure requirements: Multibiometric fusion systems may require specialized hardware and infrastructure to handle the processing and

storage demands of multiple biometric modalities. This can increase the cost of implementation and maintenance. Moreover, deploying such systems in resource-constrained environments or on mobile devices with limited processing power and storage capacity can be challenging.

3. Data availability and interoperability: Multibiometric fusion relies on the availability of multiple biometric modalities from individuals. However, obtaining data from all modalities for all individuals may not always be feasible or practical. Some individuals may not have certain biometric traits, or the quality of data from different modalities may vary. Achieving interoperability between different biometric sensors and systems can also be challenging due to differences in data formats and protocols.

1.5 Questions will Answers

The objective of this thesis is to answer the following research queries:

Research Query1: How many levels of fusion are there to fuse multiple traits?

Research Query2: What are the most popular levels of fusion?

Research Query3: What are the existing multimodal databases?

Research Query4: What are the vulnerabilities of multimodal biometric?

Research Query5: What are the Challenges in multimodal biometric systems?

Research Query6: What are the future directions in the area of multimodal biometric systems?

1.6 Literature Survey

One of the common and increasingly effective machine learning techniques used for feature selection, object classification, and object filtering processes is deep learning (DL) [7].This section provides a survey of the most current developments in the area of biometric fusion. The use of deep learning and artificial intelligence techniques in this field is particularly highlighted.

In [8] Alay and Al-Baity proposed new multimodal biometric human identification system is proposed, which is based on a deep learning algorithm for recognizing humans using biometric modalities of iris, face, and finger vein. The structure of the system is based on CNNs which extract features and classify images by softmax classifier. For fusing the CNN models, different fusion approaches were employed. The performance of the proposed system was empirically evaluated by conducting several experiments on the SDUMLA-HMT dataset. The results showed achieving an accuracy of 99.39%, with a feature level fusion approach and an accuracy of 100% with different methods of score level fusion.while our proposed model we using five different CNN , same datasets, three steps of preprocessing and achives accuracy 97.55% and 99.37% with feature and score level respectivly.

In [9] Mahmoud et al suggested a multimodal biometric identification technique to verify a person's identity using his or her iris and face features. The work used feature-level fusion utilizing a novel fusion technique.Through a series of experiments using the SDUMLA-HMT database, the suggested systems' performance was confirmed and assessed. The suggested approach has produced results with a reduced equal error rate (EER) and up to 99% recognition accuracy. while our proposed model we using five different CNN , same datasets, three steps of preprocessing and achives accuracy 97.55% and 99.37% with feature and score level respectivly.

In [10] Yadav used the biometric modalities of iris, fingerprint, and handwritten signature, a new multi-modal human identification model A proposed algorithm based on deep learning. The architecture of the system is based on convolutional neural networks (CNNs), feature and score-level fusion techniques were applied. The multimodal biometrics dataset SDUMLA-HMT was used to conduct many tests to evaluate the performance of the suggested system. The acquired results showed that accuracy of 99.11 percent with the feature-level fusion technique and an accuracy of 99.51 percent with a different approach of score-level fusion. while our proposed model we using five different CNN , same datasets, three steps of preprocessing and achives accuracy 97.55% and 99.37% with feature and score level respectivly.

In [11] Kamlaskar and Abhyankar proposed a feature-level fusion using Canonical Correlation Analysis (CCA) to combine the feature sets of a person's iris and fingerprint. The multimodal dataset 'SDUMLA-HMT' are taken into account in this experiment to evaluate the effectiveness of the proposed system. They demonstrated that the performance of the suggested technique greatly beats that of the unimodal system in terms of equal error rate (EER). a study of the correlation between images of the Left and Right Fingerprints (EER of 0.1050 percent) and the Left and Right Iris (EER of 1.4286 percent) is presented in order to take into account the impact of laterality and feature dominance of the selected modalities for the trustworthy multimodal biometric system. while our proposed model we using five different CNN , same datasets, three steps of preprocessing and achives accuracy 97.55% and 99.37% with feature and score level respectivly.

In [12] Vijayakumar constructed a multimodal biometric user identification model . The system performs feature extraction using the CNN deep learning method to accurately and error-free identify an individual. The face, iris, palm print, and finger vein were used to identify the subject, along with two different ways of score fusion. This research created a hybrid deep-learning classification approach from scratch that are appropriate for each character. For palm-print picture recognition, for instance, combining CNN with a finger vein support vector machine (SVM) delivers increased precision. Generally speaking, the SVM can classify the images well. Additionally, this research used deep learning algorithms to investigate a wider range of identification features, including DNA, signatures, and hand shapes. And it achieved a 94% accuracy rate. while our proposed model we using five different CNN , same datasets, three steps of preprocessing and achives accuracy 97.55% and 99.37% with feature and score level respectivly.

In [13] Channegowda and Prakash designed multimodal biometric models to increase the accuracy of recognizing a person. This approach makes use of a combination of physiological and behavioral biometric traits. The qualities of signature biometrics and fingerprint are integrated to create a multimodal recognition system. Biometric traits are used to derive histograms of oriented gradients (HOG) features, which are then fused at two different levels. Deep learning neural network models are then trained using these features. The outcomes of the proposed study are assessed by a variety of hidden layers and hidden neurons using Deep learning classifier and multi-level feature fusion for multi-modal biometrics. On the MCYT and SDUMLA-HMT signature biometric recognition datasets, experiments are conducted, and positive results were obtained. It achieved a 93.33% accuracy rate. while our proposed model we using five different CNN , SDUMLA-

HMT datasets, three steps of preprocessing and achives accuracy 97.55% and 99.37% with feature and score level respectivly.

In [14] Sarangi et al suggested a recognition system this system not only addresses the drawbacks of ear biometrics but also boosts the total recognition rate. It is based on the profile face and ear. the kernel discriminative common vector (KDCV) technique is utilized over the collected feature set. With the help of deep features extracted from three well-known pre-trained CNN models, namely AlexNet, GoogleNet, and VGG16, the effectiveness of the proposed model has been confirmed. Experimental results on two benchmark datasets unmistakably demonstrate that the suggested strategy outperforms individual modalities and other cutting-edge techniques in terms of performance.

Table 1.1 A Summary of Literature Review

References	**Authors**	**Biometric Traits used**	**Fusion Approach**	**used algorithms**	**datasets**	**Performance Metrics**
[8]	(Alay and Al-Baity 2020)	Iris, Face, and Finger Vein	a feature level fusion , different methods of score level fusion	CNN model (VGG-32), softmax classifier	SDUMLA-HMT	Accuracy =99.39% Accuracy =100%
[9]	(Mahmoud , Selim, and Muhi 2020)	Iris & Face	a feature level fusion	(R-HOG)	SDUMLA-HMT Database	Accuracy = 99%
[10]	(Yadav 2021)	Iris,Finger print and written signature	a feature level fusion , different methods of score level fusion	CNN model (VGG-32), softmax classifier	SDUMLA-HMT	Accuracy =99.11% Accuracy =99.51%

[11]	(Kamlaskar and Abhyankar 2021)	iris - fngerprint	a feature level fusion	canonical correlation analysis (CCA)	SDUMLA-HMT	Right Iris and Right Fingerprint images (EER of 0.2812%) and b) Right Iris and Left Fingerprint images (EER of 0.1050%)
[12]	(Vijayakumar 2021)	iris, face, finger vein, and palm print	a feature level fusion score level fusion	CNN	USM and SDUMLA-HMT	Accuracy =94%
[13]	(Channegowda a. b. and Prakash 2021)	Fingerprint - Signature	multi-level feature fusion	CNN	fingerprint from SDUMLA-HMT and signature from MCYT	Accuracy =93.33%
[14]	(Sarangi et al. 2021)	ear - profile face	a feature level fusion	CNN (AlexNet, VGG16 and GoogleNet)	side face images of the collection E (UND-E) and collection J2 (UND-J2) databases.	Accuracy =99.05

1.7 Layout of the Study

The reset of thesis has four chapters in addition to chapter one, organized as follows:

1. **Chapter Two: (Theoretical Background):** The basic functions and activities performed by multi-biometric systems and the corresponding databases are covered in this chapter. In this thesis, deep learning techniques and evaluation measures are applied.
2. **Chapter Three: (The Proposed Model):** The design and implementation of the suggested models, which include two multimodal and three single-modal biometric systems, are covered in this chapter.
3. **Chapter Four: (Experimental Results and Evaluation):** This chapter provides an explanation of the experimental findings for each stage of this work.
4. **Chapter Five: (Conclusions and Future Works):** This chapter presents the thesis' conclusions and provides a list of recommendations for future work

Chapter Two
Theoretical Background

2.1 Introduction

This chapter provides an overview of the strategies and methods used in the multimodal biometric model that has been developed. The following is how this chapter is organized: First, a summary of biometric models is given, along with descriptions of how face, iris, and fingerprint identification models work. Second, an explanation of CNN and artificial neural networks is provided, along with a discussion of the image processing techniques employed, the source database, and performance metrics.

2.2 Biometric Model

A biometric model essentially consists of a pattern recognition model that collects biometric information from a person's traits, extracts a set of characteristics for training, and then is prepared it to perform personal identification or verification. [15]**.**

2.2.1 Modules of the Biometric Model

The generic biometric model primarily consists of four primary modules that work together to perform various functions. Below is a description of each of these modules [16] .

1. Sensor module: The sensor module is responsible for capturing biometric data from individuals. It includes the hardware components such as fingerprint scanners, iris scanners, cameras, microphones, or other sensors specific to the biometric modality being used. The sensor module converts the biometric traits into digital signals or images for further processing.
2. Feature extraction module: The feature extraction module analyzes the preprocessed biometric data to extract distinctive and discriminative features. It uses algorithms and techniques specific to the biometric modality being

used. For example, in fingerprint recognition, this module may extract minutiae points, while in face recognition, it may extract facial landmarks or texture descriptors. The extracted features serve as a compact representation of the biometric data.

3. Matching module: The matching module compares the acquired biometric template with the templates stored in the database to determine the identity of the individual. It uses matching algorithms to measure the similarity or dissimilarity between the templates. The matching process may involve statistical modeling, pattern recognition techniques, or machine learning algorithms depending on the biometric modality and system requirements.
4. Database module: The database module stores and manages the biometric templates of enrolled individuals. It allows efficient storage, retrieval, and indexing of templates for matching operations. The database module may also include functionalities for template enrollment, deletion, updating, and backup.

These modules work together to form a complete biometric system.

2.2.2. The Biometric System's Phases

The training phase and the testing phase are the two primary phases:

A- Training Phase: This phase includes gathering the biometric information from the person who wants to sign up for the system and storing it in the system database. First, the biometrical attribute is scanned by a biometric sensor, which creates the initial data. A pre-processing step is then performed on the data to guarantee its validity. From this initial data, features are then taken in order to create a feature set. In order to use this feature set in the identification or verification processes later, it is kept in the system database [15] .

B- Testing Phase: In this phase, the biometric data is collected from the already enrolled person and goes through the same feature extraction and pre-processing steps to produce a feature set that will be compared to corresponding reference feature sets that are stored in the database to establish the identity of the person. Identity management functionalities come in two processes : Identification and Verification Modes [17]:

1. ***Verification Mode:*** The verification mode is a one-to-one matching process. In this mode, the person claims an identity, which is typically an Identification Number, User Name, Label, etc. The system then compares the person's feature set with the identity by looking just at the one feature set that matches it. Depending on how closely these two feature sets resemble one another, the person is either accepted or rejected.
2. ***Identification Mode:*** Since the person does not claim an identity, the identification mode is similar to verification but uses a one-to-many matching procedure to identify the person by looking for a match in the feature sets of all the enrolled users in the system database (or fails if the individual is not enrolled). Identification mode serves as a convenience or a means of preventing one person from utilizing several identities (the user is not preferring to claim an identity).

2.3 Face Identification Model

The difficulty in face recognition or identification is that the frontal view of different faces looks almost similar and the differences are quite small, this making many of pattern recognition techniques unsuccessful in distinguishing between them. There are several factors that cause variations in the face appearance such as illumination, age and pose variations (orientation of the face) which are the main three problems for the face recognition systems.

In essence, during the face recognition phase, features from the face are extracted and compared to those that the system has been trained on. This phase typically consists of a feature extraction step, a classification or matching step, and occasionally, face normalization techniques are used to normalize the face's photometric and geometrical characteristics [18].

2.4 Fingerprint Identification Model

One of the most attractive biometric technology types is the fingerprint identification system. Each person's finger has distinguishing features that can be used to identify or confirm them [13].

2.5 Iris Identification Model

Iris is one of the most interesting type of biometric technologies. It refers to the automated process of recognizing individuals based on their iris patterns. Iris recognition algorithms have demonstrated very low false match rates and very high matching efficiency in large databases. This is not entirely surprising given the complex textural pattern of the iris stroma that varies significantly across individuals, the perceived permanence of its distinguishing attributes, and its limited genetic penetrance [18]. A large-scale evaluation conducted by the National Institute of Science and Technology (NIST) has further highlighted the impressive recognition accuracy of iris recognition in operational scenarios [19]. According to a report from 2014 [8], over one billion people worldwide have had their iris images electronically enrolled in various databases across the world. This includes about one billion people in the Unique Identification Authority of India (UIDAI) program, 160 million people from the national ID program of Indonesia, and 10 million people from the US Department of Defense program. Thus, the iris is likely to play a critical role in next generation of large-scale identification systems.

2.6 Artificial Neural Networks

Artificial Neural Networks (ANNs) architectures are similar to the biological structure of the human brain's neurons. Neural networks have recently become the standard tool for solving a variety of computer vision problems [20]. One of the most basic ANN topologies is the Feed forward Neural Network. These neural networks consist of three types of layers: layers for input, output, and hidden layers. For pattern recognition tasks, external data is received by an input layer. An output layer offers the answer to the issue. A hidden layer that links the other layers processes data entries and map the inputs to the outputs. The more hidden layers a network has, the deeper the network is considered. Fig 2.1 represent the structure of a single neuron in the ANN architecture

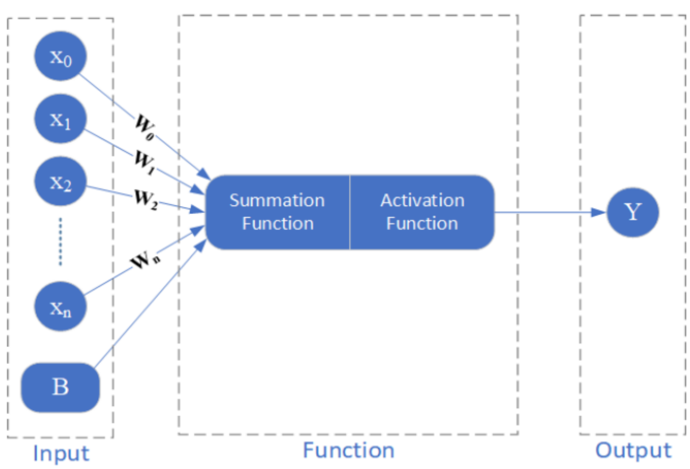


Figure2.2 Structure of a single Neuron [21]

As shown in Fig 2.1, a single neuron receives a number of inputs from nodes in the previous layer. All inputs are weighted differently by multiplying each input by a learnable parameter called Weight. Then, the neuron sums up all of its input with another learnable parameter called Bias [22].

$$Summation\ Function: \quad \xi = \sum \ xi(W_i + B_n) \qquad i = 0 - n \qquad (2\text{-}1)$$

$$Activation\ Function: Y = (\xi) \qquad (2\text{-}2)$$

$where: xi = input\ acquired\ from\ previous\ layer$

$Wi = learnable\ weight$

$B = learnable\ bias$

$n = number\ of\ inputs$

$\xi = summation\ output$

$Y = neuron\ output$

2.7 Deep Learning

In the traditional machine learning approach, the system receives inputs from specified features. In other words, just the most crucial and important elements are chosen or created [23]. The engineer or programmer performs this manually. that means a software engineer would have to select the relevant features in a more traditional Machine Learning algorithm (manually choose features and a classifier) because traditional Machine Learning algorithms have a rather simple structure, such as linear regression or a decision tree, while Deep Learning algorithms require much less human intervention [24]. The features are extracted automatically and the algorithm learns from its own errors. However, the best outcomes are not usually the result of doing this. Considering that there are several phases and numerous methods for dealing with them. Furthermore, the greatest outcomes are not always achieved by manually selecting and developing features.[24]. Fig 2.2 illustrates the key distinction between ML (machine learning) and DL (deep learning).

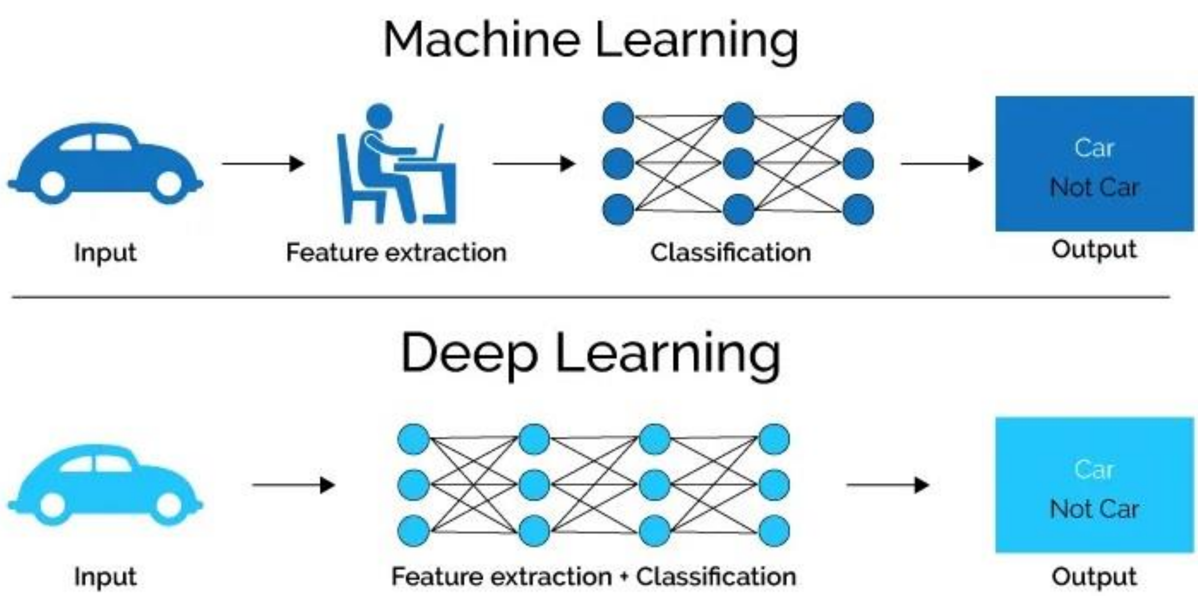


Figure 2.2 Basic Distinctions Between Deep Learning and Machine Learning [25]

Briefly, Deep learning, machine learning, and artificial intelligence are related concepts but differ in their scope and functionality. Table 2.1 explains a comparison between these terms:

Table 2.1 A comparison Between Deep Learning and Machine Learning and Artificial intelligence [23]

Artificial Intelligence	Machine Learning	Deep Learning
AI refers to the broader field of computer science that aims to create intelligent machines capable of performing tasks that typically require human intelligence	ML is a subset of AI that focuses on developing algorithms and statistical models that enable computer systems to learn from data without being explicitly programmed	DL is a subfield of machine learning that specifically utilizes artificial neural networks inspired by the structure and function of the human brain.
It involves the development of algorithms or systems that can perceive, reason, learn, and make decisions. AI encompasses various subfields, including machine learning and deep learning.	ML algorithms learn patterns and relationships in large datasets, allowing them to make predictions or take actions based on new, unseen data.	DL models, known as deep neural networks, consist of multiple layers of interconnected nodes (neurons) that process and transform input data. DL algorithms learn hierarchical representations of data, automatically extracting features at different levels of abstraction

artificial intelligence is the overarching field	traditional ML algorithms often require manual feature engineering	DL algorithms can automatically learn relevant features from raw data, reducing the need for manual feature extraction.
artificial intelligence is the overarching field	ML algorithms can also achieve high performance but may require more effort in feature engineering.	its ability to learn hierarchical representations, has achieved state-of-the-art performance in various domains, especially those with complex and large-scale datasets
artificial intelligence is the overarching field	ML models are often more interpretable, as the features and decision-making processes are explicitly defined	DL models, with their complex architectures, can be more challenging to interpret, often referred to as "black boxes."
artificial intelligence is the overarching field	ML has achieved success and requires less data and computational resources compared to DL techniques.	DL has achieved remarkable success but requires more data and computational resources compared to traditional ML techniques.

2.7.1 Deep Neural Networks (DNN)

DNN is ANN with several hidden layers in theory. One of the ANN structures that is frequently utilized for DNN is the MLP. It is nearly impossible to successfully train more than a few hidden layers since neural networks are made up of layers of coupled neurons. A network can have dozens or even millions of weights, hence the DNN needs a lot of data to be fed into the training phases as well as incredibly long computation times [26] [27].

Fig 2.3 represents a DNN architecture with (5) input nodes, (4) hidden layers with (7) neurons, and (4) output nodes, and simple neural network.

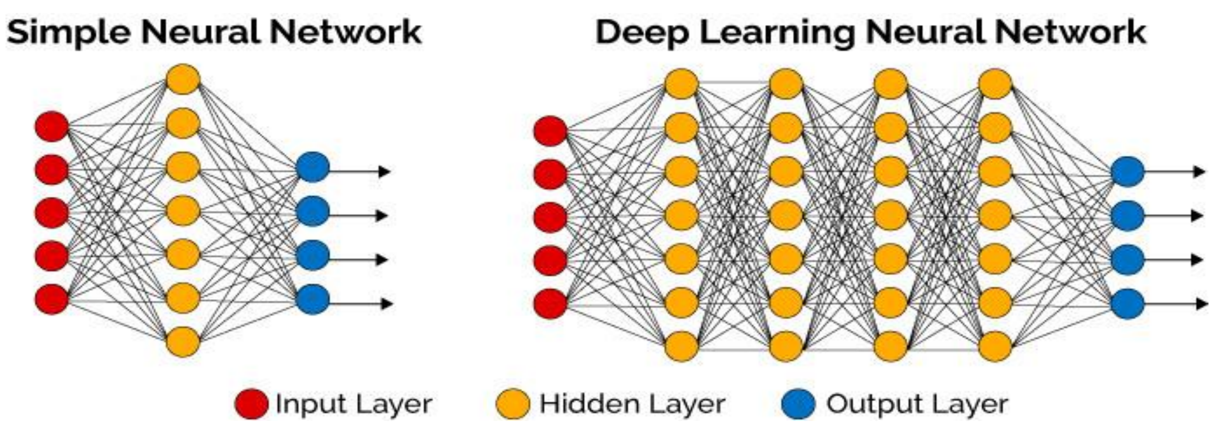


Figure2.3 Architecture of Deep Neural Network [21]

A hidden layer can have any number of neurons that connect to one or more nodes from the previous layer. A layer can be Fully-Connected If its neurons connect to all outputs of the last layer. While ANN architecture can have only one layer, DNN consist of any number of hidden layers. This helps DNN architectures to go deeper than ANN in any defined task. Deeper means that a network can solve more complicated and advanced problem [28].

Fig 2.4 represents a DNN architecture with (3) input nodes, (4) hidden layers with (5) or (7) neurons, and a single output node. Other types of DL architectures differ in the neuron functionality or the nodes' connection scheme. Fig 2.3 shows different types of neural networks architectures. Input layers consist of nodes equal to the number of input elements. This means an image requires as many nodes as the number of pixels in an image array. Whereas; the number of output nodes indicates the number of classification categories in a classification task. Determining the number of hidden layers and neurons is problematic as it requires further knowledge and many trials [29].

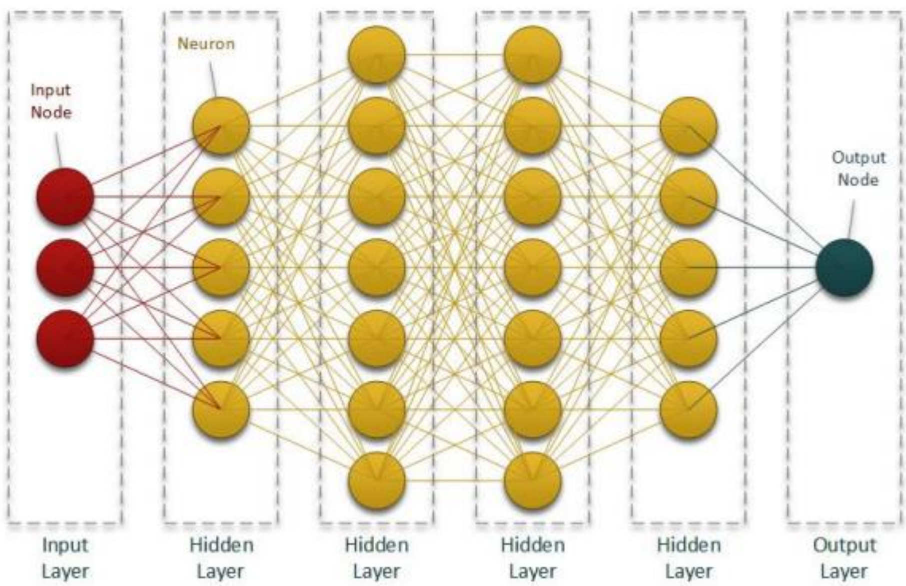


Figure2.4 Feed Forward Deep Neural Network Architecture [30]

2.7.2 Convolutional Neural Networks

Convolutional Neural Network (CNN or ConvNet) is a common type of Neural Network commonly used in computer vision applications such as image classification and segmentation [22]. CNNs are similar to traditional neural networks in that they consist of neurons with learnable weights and biases. But the neurons in a CNN hidden layer are connected efficiently to extract features and patterns from images. If a 2D image had $(N \times M)$ number of pixels, a feed forward DNN architecture would require $(N \times M)$ learnable weights per a single neuron.

CNN shares a defined number of learnable weights across all input elements. Due to this, large-scale network designs are much easier with CNNs than other types of DL architectures. CNNs are considered to be space invariant, that is, classification ability is not influenced by features' location within an input image. This is a desirable characteristic for pattern recognition since a CNN can detect and extract a feature regardless of its location [29]. An example of CNN architecture is illustrated in Fig 2.5.

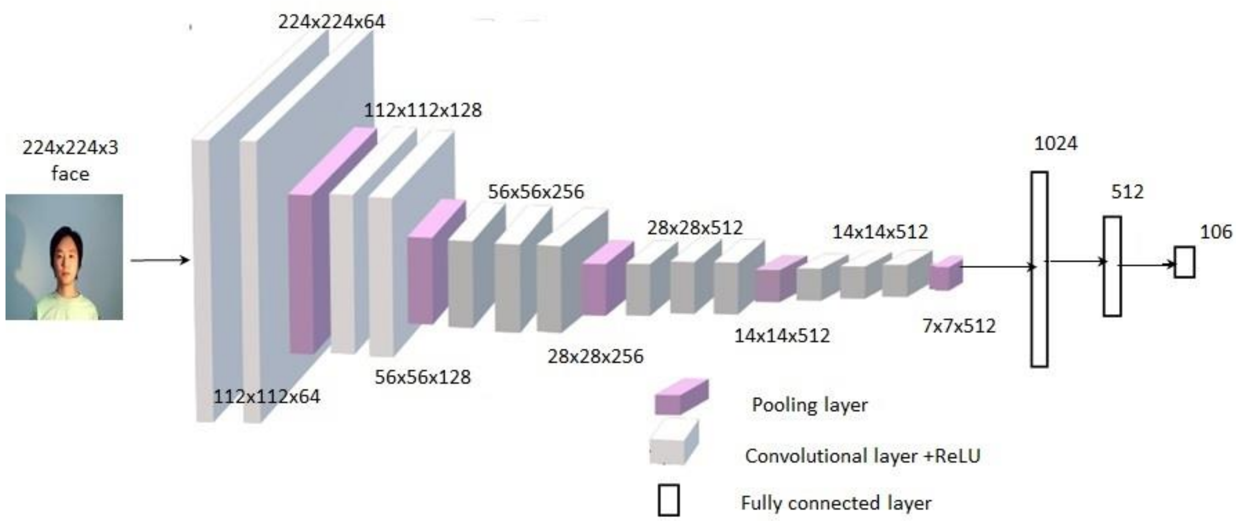


Figure2.5 Architecture of Convolutional Neural Network [22]

A commonly used type of a CNN architecture, consists of numerous convolution operation proceeded by several Down-Sampling operations. CNN architectures consist of various types of layers, or so-called the layers or building blocks. Each layer type has a different functionality and use case scenario [31].

2.7.2.1 Convolution Layer

It is the most significant layer that performs the essential convolution operation in CNN architecture. Convolution layers have one or more arrays of learnable weights called filters, or kernels, whose values need to be learned. Each kernel, in a convolution operation, is convolved with the input to compute the output. In other words, each kernel is slid across the input and the summation of the dot products between the input and the kernel is computed at every single position.

Outputs of the convolution operation are called feature maps; or activation maps. The number of feature maps is equal to the number of kernels as each kernel would output a single feature map. Trained kernels within several consecutive convolution layers can extract hard features efficiently [22] . Convolution operation of a (3×3) kernel is illustrated in Fig 2.6.

Two concepts are important can impact the shape of the feature map when performing a convolution operation:

- Stride: is the number of steps a kernel moves when sliding across the input map. The stride of two means that the filter will slide across two elements at a time, skipping one when moving in a direction.

- Padding: is the process of adding values to the outside boundary of the input feature map. When aligning the center of a kernel to a corner of the input, (2) kernel edges would go beyond the input border where there are no values. Hence, the filter will start with the pixel next to the input borders if no padding existed. By default, padding values are “0”

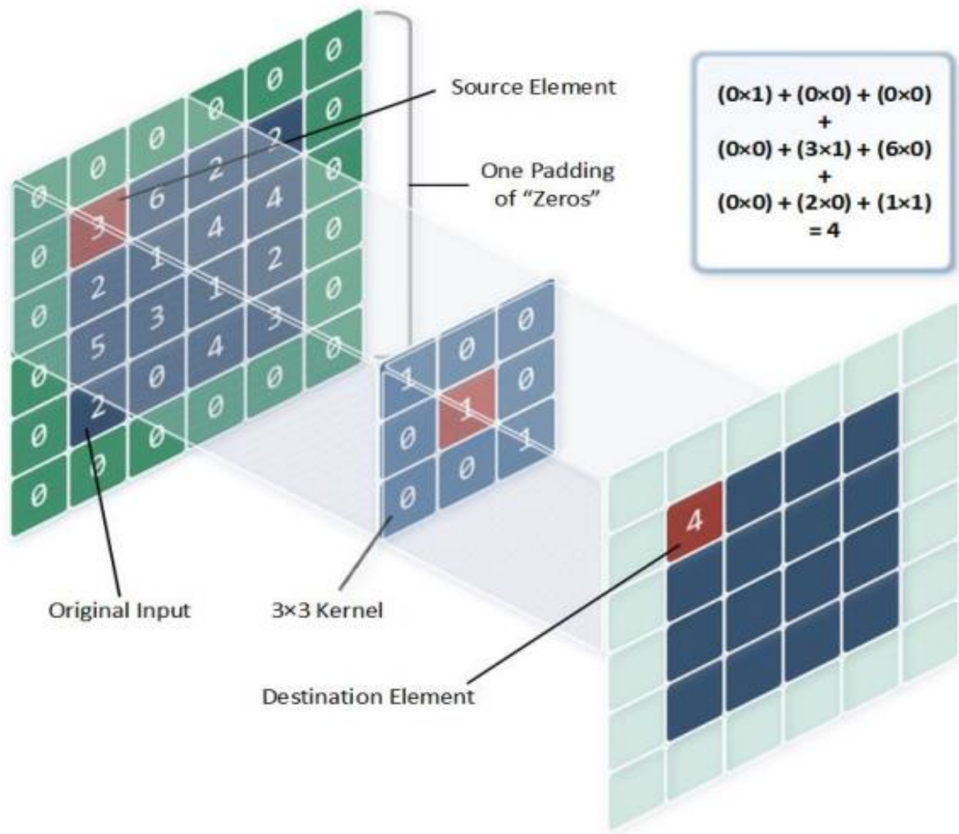


Figure2.6 2D Convolution Operation [22]

2.7.2.2 Pooling Layer

The pooling layer, also called the down-sampling layer, usually lies between a set of convolutional layers. Pooling layers take a small patch of its input and sub-sample it to produce a single output. Pooling operation takes only one value of the selected canvas, thus reducing the dimensionality of the feature maps. Convolution layers could perform a pooling operation by themselves if they moved more than one step at a time, skipping some pixels in-between steps. Due to pooling operations, some information is lost; however, this is a necessary step to reduce the complexity of the network.

Global Pooling layers perform a more extreme dimensionality reduction than regular pooling layers. Global pooling layers do not down-sample but rather reduce the dimensionality of the input feature map to a single element. There are two types of pooling layers in general: max-pooling and average pooling. Fig 2.7 shows an example difference between pooling layers.

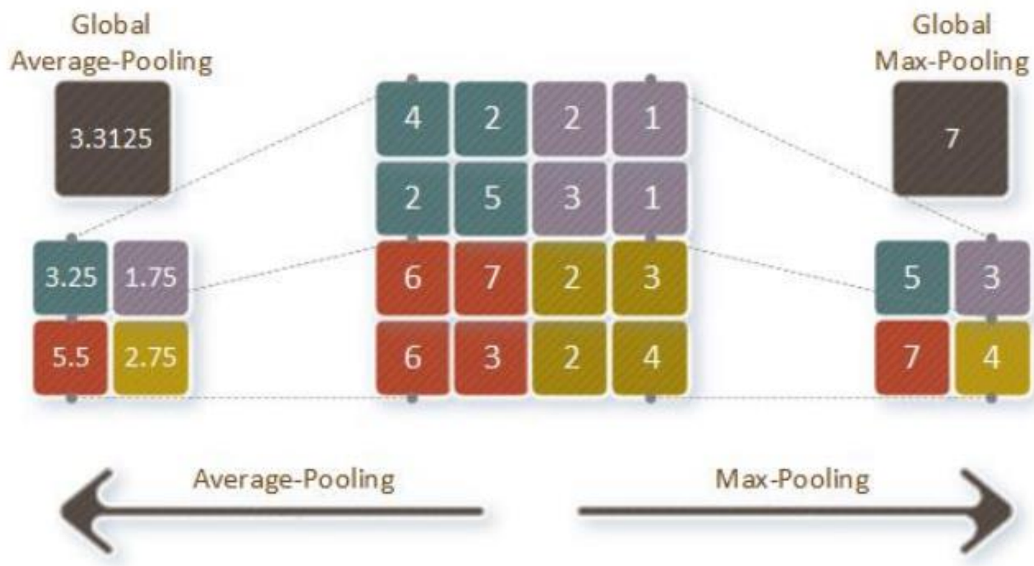


Figure 2.7 Max-pooling Operation [22]

2.7.2.3 Activation Layer

The output of the preceding layer, which is often a convolution layer, is subjected to non-linearity in activation layers. By reducing pointless values, this non-linearity enables the network to train more quickly. In neural networks, the activation function decides whether the neuron will be activated or not based on the summation function result. The same concept applies to CNN; the activation layer defines the elements of the input feature map. Several activation functions can be used like Sigmoid, Tanh, ReLU, SoftMax, etc. Fig 2.8 shows two different activation function curves.

$$Sigmoid: (x) = \frac{1}{1 + e^{-xi}} \qquad (2\text{-}3)$$

$$Tanh:(\mathrm{x}) = \frac{2}{1 + e^{-2xi}} - 1 \qquad (2\text{-}4)$$

$$ReLU: (x) = argmax\ (0, xi) \qquad (2\text{-}5)$$

$$SoftMax: (x) = \frac{e^{xi}}{\sum_{j=1}^{k} e^{xj}} \qquad (2\text{-}6)$$

$Where: xi = input\ feature\ map\ element\ (i)$

$argmax = maximum\ between\ two\ values.$

$K = number\ of\ classes\ in\ the\ feature\ map.$

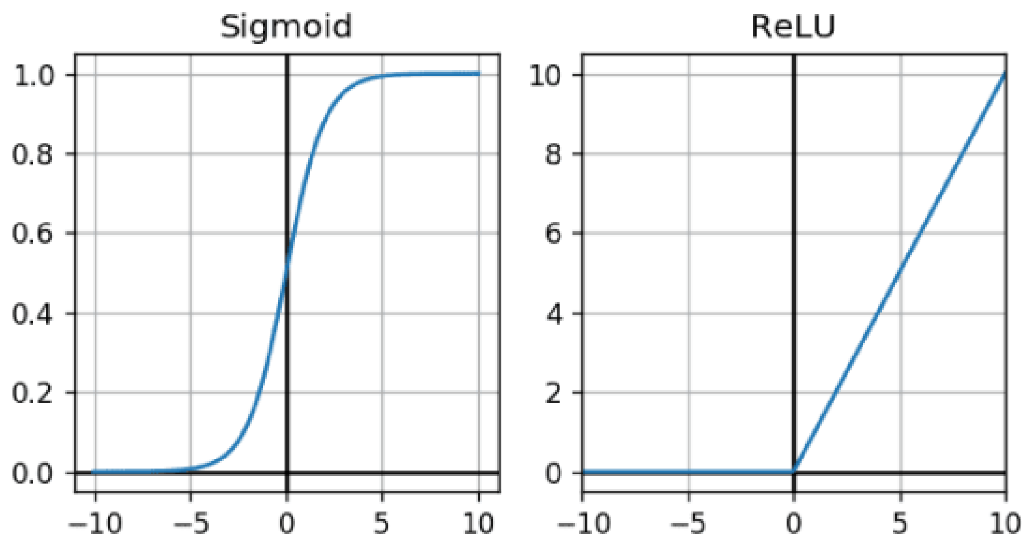


Figure2.8 Sigmoid and ReLU plots [25]

2.7.2.4 Normalization Layer

The normalization layer standardizes the inputs to a layer by subtracting the mean and dividing by the standard deviation. Normalization has the effect of stabilizing the learning process and dramatically reducing the training time required to train deep networks. Based on the dimension of the normalization; Normalization layers are categorized into: batch normalization, layer normalization, instance normalization, and group normalization.

$$normalization: X` = \frac{X - \mu}{\sigma} \quad (2\text{-}7)$$

$where: X` = normalized\ array$

$\mu = mean\ \sigma = standard\ deviation$

2.7.2.5 Deconvolution Layer

Deconvolution is a mathematical operation that works the opposite way of convolution and pooling operations. Instead of sliding the kernel across the input feature map, each element of the feature map slides across that kernel.As a result, the input feature map is up-sampled by the kernel size. Deconvolution layers are substantial in the image segmentation field. Fig 2.9 illustrate a deconvolution operation.

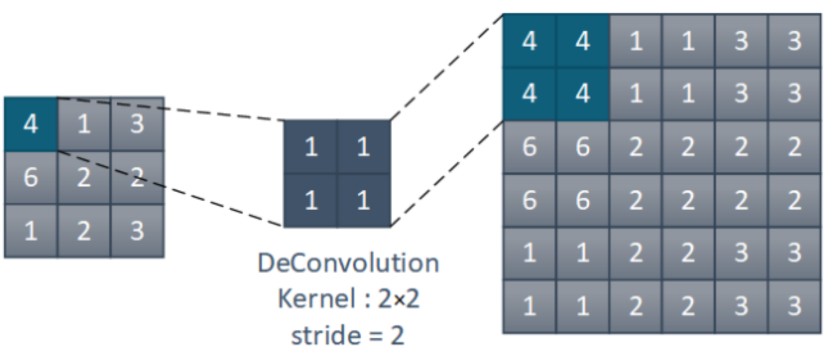


Figure 2.9 Deconvolution Operation with Stride of (2) [25]

2.7.2.6 Dropout layer

In neural networks, dropout layers switch a defined percentage of the network's neurons at random in the training phase. When neurons are switched off, the incoming and outgoing connections to those neurons drop as well. The dropout percentage must be lower than (50%) to avoid dropping out more than half of the model. The dropout layer functions only during the training phase. Dropout has the effect of making the training process noisy, forcing non-dropped neurons to take more responsibility. This concept prevents learnable parameters from settling on unwanted points. The dropout layer can be implemented as follows:

Consider a neural network with L hidden layers. Let l ∈ {1, . . . , L} index the hidden layers of the network. Let y l denotes the vector of outputs from layer l. The dropout equation is the following

$$\boldsymbol{r_j^l} = \sim\mathbf{Bernoulli}(\mathbf{p}) \qquad (2\text{-}8)$$

$$\boldsymbol{y}^{\sim\boldsymbol{l}} = \boldsymbol{r^l} \odot \boldsymbol{y}^{\ \boldsymbol{l}} \qquad (2\text{-}9)$$

Where r^l is a vector of independent Bernoulli random variables, each of which has probability p of being 1, y^l the outputs of layer l, $\odot$ element-wise product, and $y^{\sim l}$ is thinned output that will be as input x for next layer l+1. Fig 2.10 illustrate a dropout operation.

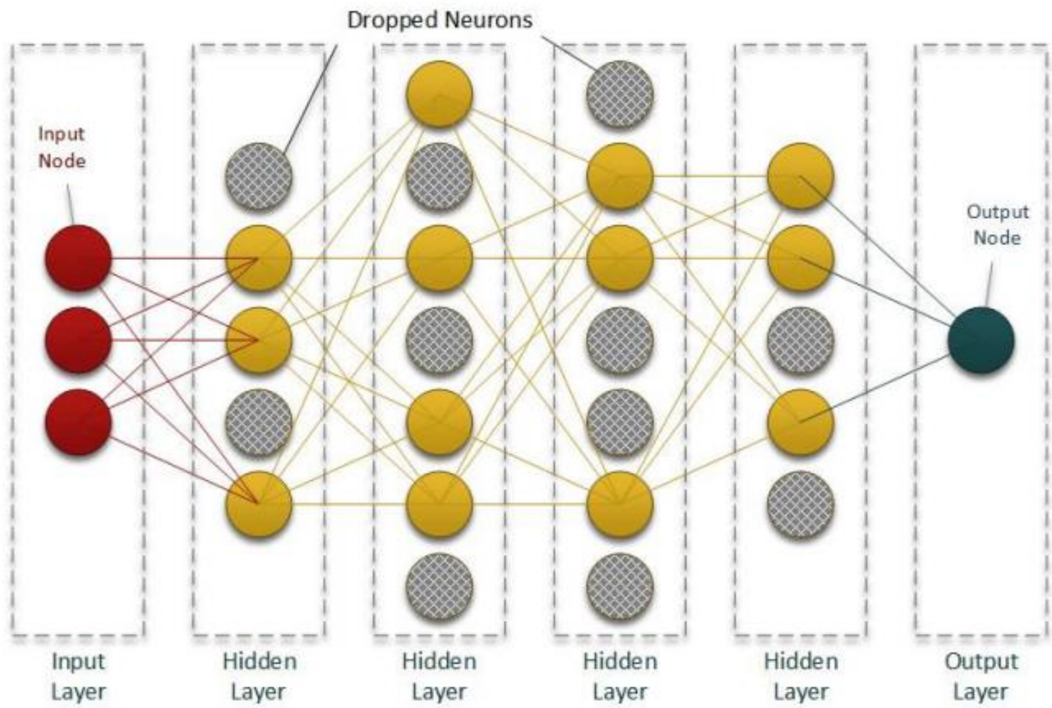


Figure 2.10 Dropout Operation [7]

2.7.2.7 Fully-Connected Layer

The Fully-Connected layer is similar to the feedforward ANN architecture that has only one hidden layer. Fully-connected indicates that all the neurons of these layers connect to all input map elements of this layer. Each neuron produces a single output making the number of outputs equal to the number of neurons. Usually, fully-connected falls at the end of a CNN architecture due to their classification ability. Besides, due to their fully-connected nature, these layers require an enormous number of learnable parameters. Placing the fully connected layers after several dimensionality reduction operations, like pooling, reduces the number of learnable parameters needed.

2.7.2.8 Loss Function

In the CNN model training, a loss function is performed in the output layer to calculate the predicted error created across the forward propagation pass. This error reveals the difference between the actual predicted and desired output. The main goal of the network training is to minimize the loss function, also known as the cost function, by updating the learnable parameters iteratively. Choosing a loss function for specific problem is not straightforward and may depend on the task in hand. Several loss functions can be used such as Cross-Entropy, and Mean Squared Error (MSE) [22] . Cross-entropy equation are described below:

$$\boldsymbol{z_j = -\frac{1}{n}\sum_{i=1}^{N} y^i \ln g\,(z^i) + (1 - y^i) \ln\left(1 - g(z^i)\right)} \quad (2\text{-}11)$$

Where n is number of classes, y^i is the desired output for the i'th class . $g\left(z^i\right)$ is the estimated probability of class i

2.7.2.9 Forward Propagation of Subsampling layer

Feed forward operation of subsampling layer is straight forward, since the input feature-map is divided into none-overlapping sub-regions, mostly the size is 2×2 and each of these produces one output. Another possibility is to have overlapping but using larger size of sub-regions [32]. The max-pooling subsampling operation is defined as in equation (2.12):

$$\boldsymbol{x^l_{q.u.v} = max_{\mathbf{ab}\,\in(0\mathbf{s}-1)}(y^l_{\mathbf{q}.(\mathbf{i}+\mathbf{a}).(\mathbf{i}+\mathbf{b})})} \quad (2\text{-}12)$$

where $\boldsymbol{q} \in \boldsymbol{F}\ \boldsymbol{l}$ feature set, **s** is the size of pooling sub-region

2.8 Image Processing Techniques

This section discusses the image processing methods that were employed in this thesis to enhance and preprocess the source images.

2.8.1 Bilinear Interpolation Method for Image Resizing

Bilinear interpolation is a commonly used method for image resizing. It is a technique that estimates new pixel values based on the surrounding pixels in the original image. Bilinear interpolation considers the four nearest pixels to the desired location and calculates the weighted average of their values to determine the new pixel value [33].

Here's how the bilinear interpolation method works for image resizing:

1. Determine the target size: Decide on the desired dimensions (width and height) for the resized image.
2. Calculate the scaling factors: Calculate the scaling factors for both the width and height by dividing the target size by the original size. For example, if the original image has dimensions of 100x100 pixels, and you want to resize it to 200x200 pixels, the scaling factors would be 2 for both width and height.
3. Iterate over each pixel in the resized image: For each pixel in the resized image, perform the following steps:
   a. Calculate the corresponding position in the original image: Divide the coordinates of the pixel in the resized image by the scaling factors to find the corresponding position in the original image. For example, if the current pixel coordinates in the resized image are (50, 50) and the scaling factors are 2, the corresponding position in the original image would be (25, 25).

b. Determine the four nearest pixels: Identify the four nearest pixels around the corresponding position in the original image. These pixels will be used for bilinear interpolation.

c. Perform bilinear interpolation: Calculate the weighted average of the pixel values of the four nearest pixels based on their distances from the corresponding position. The closer a pixel is to the corresponding position, the higher its weight in the average. This weighted average determines the new pixel value for the resized image.

4. Repeat the process for all pixels in the resized image: Iterate over each pixel in the resized image and perform bilinear interpolation to determine their new values.

By using bilinear interpolation, the resized image is generated by smoothly blending the colors and intensities of the original pixels, resulting in a more visually pleasing and natural-looking image [34].

2.8.2 Data Augmentation

Data augmentation is a technique used in computer vision and machine learning to artificially increase the size and diversity of a training dataset by applying various transformations to the original images. This helps improve the performance and generalization of models by exposing them to a wider range of variations in the data [35]. Here are some common data augmentation techniques applied to images:

1. Rotation: Rotate the image by a certain angle, such as 90 degrees, 180 degrees, or a random angle within a specified range.
2. Flip: Flip the image horizontally or vertically to create a mirror image.
3. Scaling and Resizing: Scale the image up or down by a certain factor.

Resize the image to a specific width and height.

4. Translation: Shift the image horizontally or vertically by a certain number of pixels.
5. Shear: Apply a shearing transformation to the image, which slants the image along a particular axis.
6. Zoom: Zoom in or out of the image by either cropping or padding the image.
7. Noise Injection: Add random noise to the image, such as Gaussian noise or salt-and-pepper noise.
8. Color Jitter: Modify the color of the image by adjusting brightness, contrast, saturation, or hue.
9. Elastic Distortion: Apply elastic deformations to the image, mimicking the effects of stretching and contracting.
10. Random Cropping: Randomly crop a portion of the image to focus on specific regions.
11. Random Erasing: Randomly erase a rectangular portion of the image, simulating occlusions or missing data.

These are just a few examples of data augmentation techniques for images. The specific combination and parameters of augmentation techniques depend on the specific task, dataset, and desired variations. By applying these transformations to the original images, you can generate a larger and more diverse dataset for training machine learning models, ultimately improving their robustness and performance.

2.8.3. Data Normalization

Dataset normalization is a preprocessing step in machine learning that aims to standardize the feature values of a dataset. The goal of normalization is to bring all features onto a similar scale, which can improve the performance and convergence of machine learning models [36]. Here are some common techniques for dataset normalization:

1. Min-Max Scaling (Normalization)
2. Standardization (Z-score normalization)
3. Robust Scaling
4. Unit Vector Scaling

These normalization techniques ensure that each feature has a similar scale, avoiding the dominance of certain features with larger values and enabling the model to learn from all features equally. The choice of normalization technique depends on the specific characteristics of the dataset and the requirements of the deep learning algorithm being used.

2.9. Different Types of CNN Models

There are numerous CNN architectures with various convolutional, pooling, ReLU, fully connected, and custom layers. Below are descriptions of the most popular architectures.

2.9.1 LeNet-5

The first effective implementation of CNN architectures was LeNet, which Yann LeCun suggested in 1989[32]. The architecture was improved to LeNet5 and successfully used to recognize handwritten digits. LeNet-5 has a straightforward architecture made up of three fully linked layers and two convolutional layers. The final fully connected layer included 10 neurons, one for every digit class probability, and the incoming training images were (32x32) in resolution. LeNet-5 contains (60,850) learnable parameters in total. LeNet-5 architecture is shown in Fig 2.11

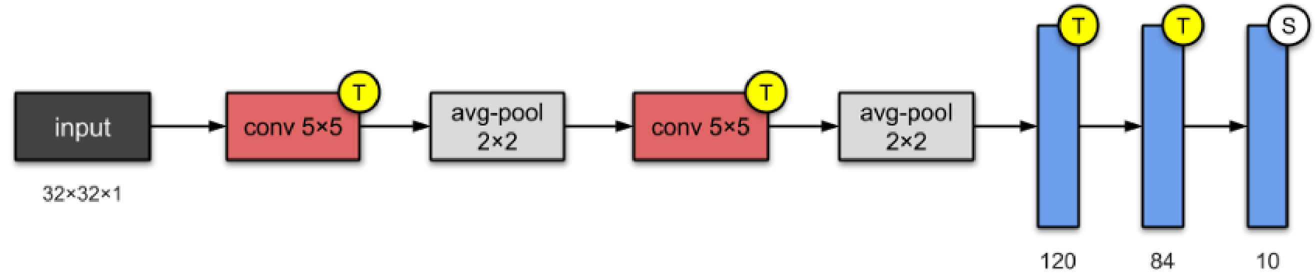


Figure 2.11 LeNet-5 Architecture [22]

2.9.2 AlexNet

AlexNet architecture was designed by Krizhevsky et al. [37] in 2012. AlexNet was the best CNN architecture at the time and includes 61 million learnable parameters. AlexNet contains three fully connected layers and five convolutional layers, making it deeper than LeNet. The AlexNet architecture is shown in Fig 2.12.

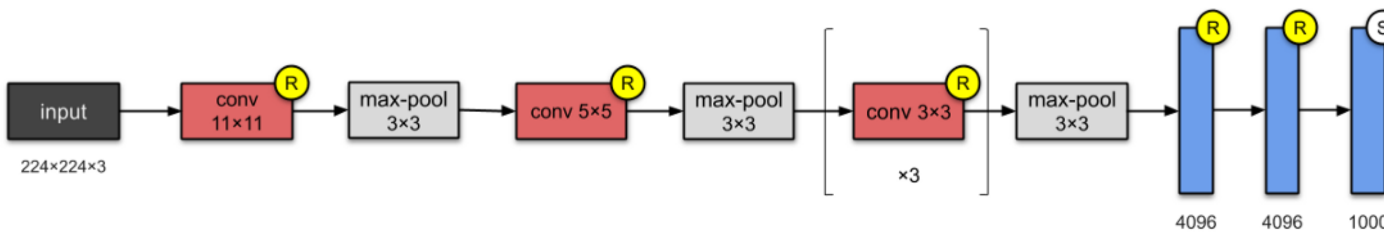


Figure 2.12 AlexNet Architecture [22]

As the first person to use the ReLU activation function, Krizhevsky's network trained far more quickly. Max-pooling was used by Krizhevsky to carry out the down-sampling procedure and a (50%) dropout layer to aid in the training process, both of which have shown to be quite successful.

2.9.3 Visual Geometry Group (VGG16)

VGG16 was designed at Visual Geometry Group (VGG) by Simonyan and Zisserman.[38] in 2014 using a more complex configuration of AlexNet . In comparison to AlexNet, VGG has a smaller (3x3) convolution kernel and (2x2) pooling kernel. VGG16's architecture typically comprises of (3) fully-connected layers, ReLU activation, and max-pooling layers, followed by (13) convolution layers.

More than twice as many learnable parameters as AlexNet are available in VGG16 (138 million). Compared to VGG16, VGG19 is a deeper model with more convolutions. Fig 2.13 provides an illustration of the VGG-16 architecture diagram.

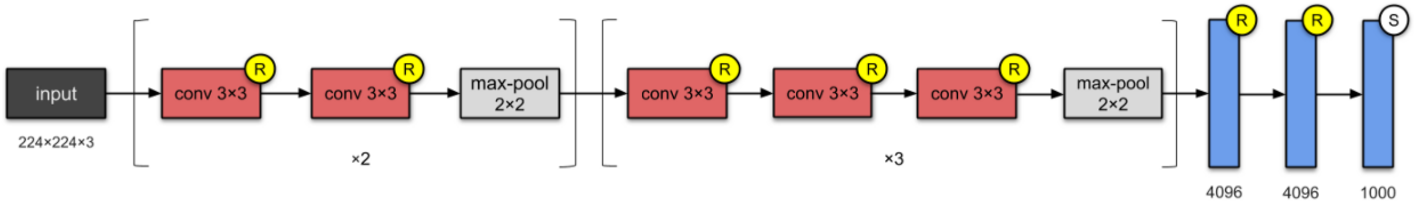


Figure 2.13 VGG16 Architecture [22]

2.9.4 GoogleNet

The architecture of GoogleNet (Inception-v1) designed by Szegedy [39], uses a module from Inception to cut down on the number of parameters. The inception module's concept is that it goes deeper and wider rather than just deeper. In other words, it employs a number of concurrent convolution layers with pooling steps to process a particular input. Feature maps are combined at the conclusion of the inception module's convolution layers. The number of levels in the GoogleNet architecture was also raised to 22.

However, AlexNet uses 12 times more learnable parameters than (5) million, which results in substantially higher accuracy. For a drastic dimensionality reduction, GoogleNet substituted one fully linked layer with a (7x7) average pooling layer [46]. GoogleNet uses parallel lines of convolutions with various filters (1x1), (3x3), and (5x5), as seen in Fig 2.14, followed by concatenation. Since it has a stride value of, the max-pooling layer does not reduce the dimensionality (1). Simply the greatest activation value is repeated over a (3x3) region. Additionally, the (1x1) convolution layers are used to increase non-linearity and decrease the amount of feature mappings.

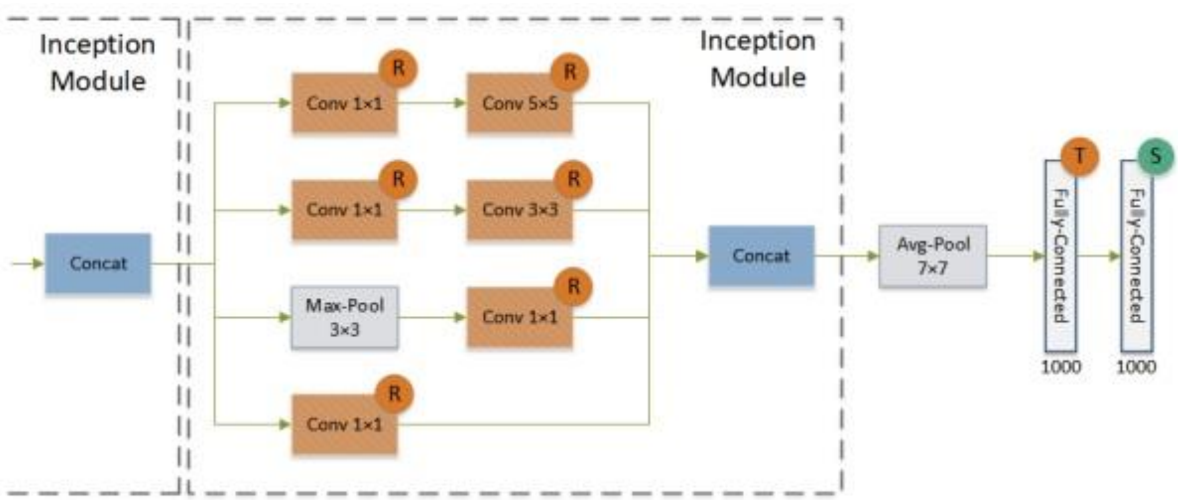


Figure 2.14 GoogleNet Module [22]

2.9.5 Residual Neural Network

Researchers have increased the number and depth of network layers, as evidenced in the aforementioned CNNs. Although deeper networks perform better, accuracy might get saturated or even decline as a result. The network's inability to learn the weight of the first layer’s results in accuracy decrease. In other words, a sufficiently deep CNN architecture may perform worse with additional layers added to it. Kaiming He et al. [40] developed Residual Network (ResNet), which introduced a novel idea. This novel idea clarified how deeper models learn via so-called bypass lines (Skip-Connections). Using skip-connections to create a path for the network to train a first layer of the network, he solved the issue with the deeper networks.

Fig 2.15 demonstrates the ResNet identity building blocks. Without degrading, ResNet can contain up to (152) layers overall. In the convolution layers, a stride value of (2) performs the max-pooling operation.

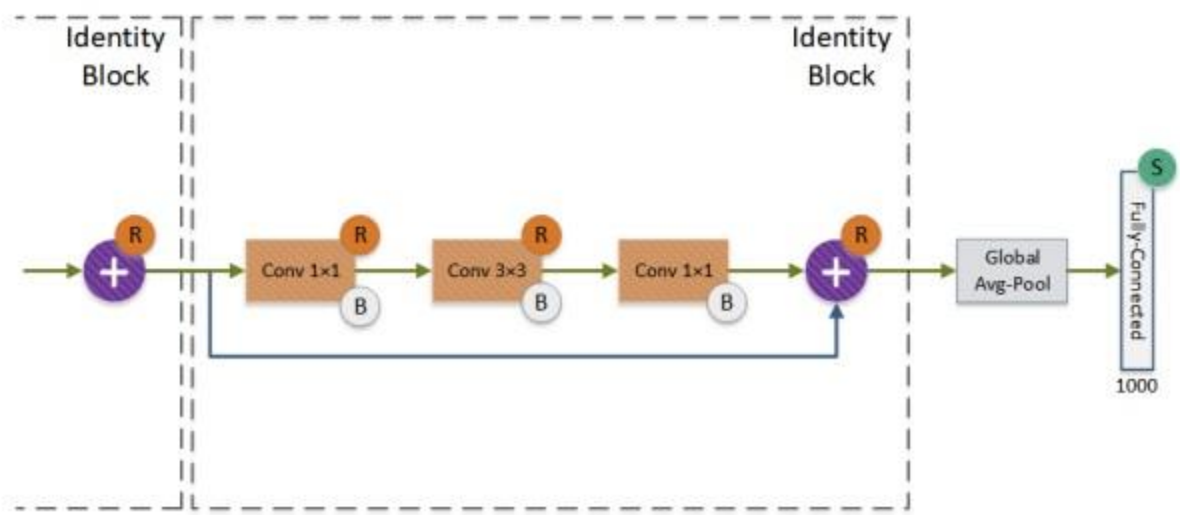


Figure 2.15 ResNet Architecture [22]

2.9.6 Mobile Net

A group of Google engineers announced the MobileNet model at CVPR 2017 [41] it features 27 convolutional layers, including 13 depth wise convolutions, one average pool layer, one fully connected layer, and one softmax layer. While lesser variants of MobileNet have 1.32 million parameters, the mainstream MobileNet model has 4.2 million parameters. The architecture of the mobile net is shown in Fig 2.16.

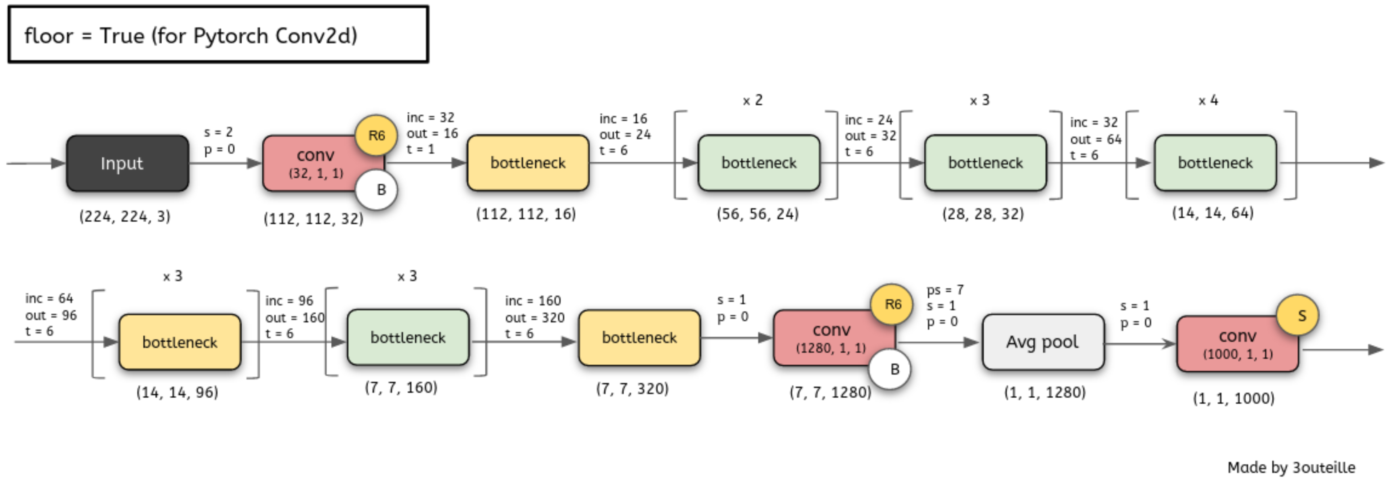


Figure 2.16 Mobilenet Architecture [22]

2.9.7 DenseNet

In 2016, Huang et al. [42] an idea from Facebook proposed DenseNet, connects every layer of a CNN to every other layer in a feed-forward manner. According to the authors, using densely connected topologies has various benefits, including "lowering the number of parameters, strengthening feature propagation, encouraging feature reuse, and relieving the vanishing-gradient problem." The Appendix contains a full description of DenseNet-201's architecture. In this study, we build the CNN Features for the iris detection task by extracting the results of a predetermined number of dense layers (15).However, we have simply selected the preceding architectures to demonstrate how well pre-trained CNNs perform on the iris identification problem. The architecture of the DenseNet is shown in Fig 2.17.

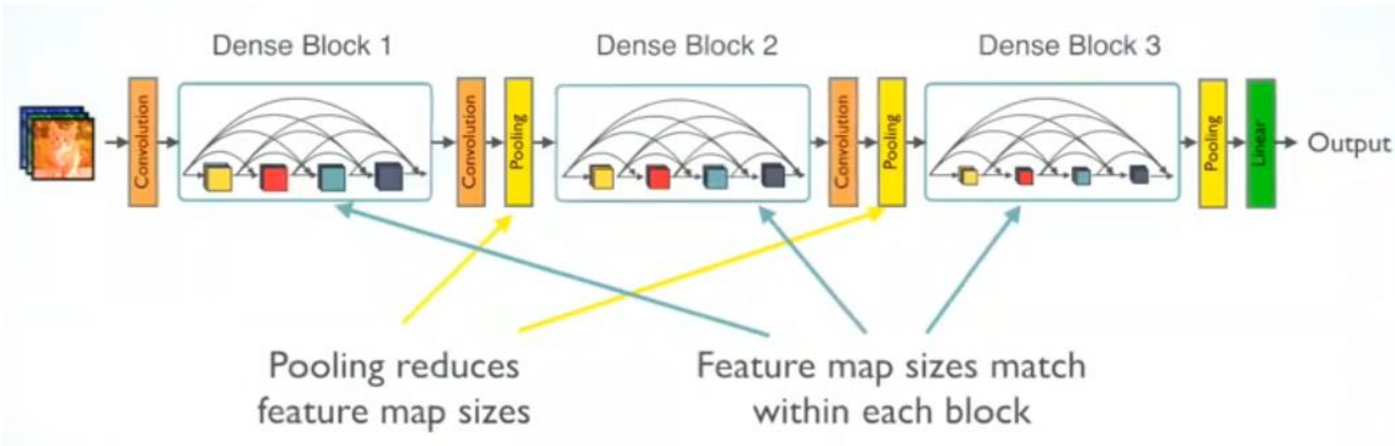


Figure 2.17 Densenet Architecture [22]

Table 2.2 A comparison Between (VGG16, ReseNet50 , MobilevNet , DenseNet , GoogleNet)

	GoogleNet	**DenseNet**	**MobilevNet**	**ReseNet50**	**VGG16**
advantages	Efficient use of parameters by employing the Inception module, which reduces computational cost. Good accuracy on image classification tasks. Deep architecture with multiple layers. Introduced the concept of auxiliary classifiers to combat the vanishing gradient problem.	Utilizes dense connections, allowing each layer to directly access the outputs of all preceding layers. Promotes feature reuse, reduces the number of parameters, and encourages better gradient flow. Addresses the vanishing gradient problem. Achieves strong performance even with fewer layers.	Specifically designed for mobile and embedded devices with limited computational resources. Utilizes depthwise separable convolutions, reducing the number of parameters and computational cost. Achieves a good trade-off between accuracy and model size. Efficiently runs on devices with low power consumption and limited memory.	Introduced residual connections, allowing the network to learn residual functions. Solves the vanishing gradient problem and enables training of very deep networks (100+ layers). Better gradient flow and improved network convergence. State-of-the-art performance on various image recognition tasks.	Simple and straightforward architecture with small 3x3 filters. Achieved excellent performance on image classification tasks. Easy to understand and implement. Suitable for transfer learning.

disadvantages	GoogleNet is more complex compared to some other models ,deep architecture and multiple layers increase the memory requirements during training ,Training GoogleNet can be computationally expensive and time-consuming due to its large number of parameters and deep structure.	Increased memory requirements ,Computational complexity ,Possible overfitting	MobileNet sacrifices a bit of accuracy compared to larger architectures due to its reduced representation power. It may not be the most suitable choice for tasks requiring fine-grained spatial information or precise localization.	Increased memory requirements ,Computational complexity,larger model size compared to some other architectures	High computational and memory requirements,Lack of efficiency: The use of small filters (3x3) results in a large number of parameters, which can make VGG16 less computationally efficient,Prone to overfitting
Primary Parameters	The primary parameter of GoogleNet is the depth, which refers to the number of layers in the network. It introduced the concept of "Inception modules," which consist of multiple parallel convolutional layers with different filter sizes.	The primary parameter of DenseNet is the growth rate, which determines the number of feature maps added to each layer in the dense block. The depth of the network is also a primary parameter that affects the overall model size and complexity.	The primary parameters of MobileNet are the width multiplier and the resolution multiplier. The width multiplier scales the number of channels in each layer, and the resolution multiplier scales the input resolution.	The primary parameter of ResNet is the depth, which refers to the number of layers in the network. ResNet introduced residual connections, allowing the network to learn residual mappings.	The primary parameter of VGG16 is the depth, which refers to the number of convolutional layers in the network (16 layers in this case). VGG16 uses small 3x3 filters throughout the network.
secondary parameters	Learning rate, regularization techniques (e.g., dropout, L1/L2 regularization), optimization algorithm, and batch size are common secondary parameters that can be adjusted for optimal training.	Learning rate, regularization techniques (e.g., dropout, L1/L2 regularization), optimization algorithm, batch size, and compression factor (for reducing the number of feature maps) are common secondary parameters.	Learning rate, regularization techniques (e.g., dropout, L1/L2 regularization), optimization algorithm, and batch size are common secondary parameters.	Learning rate, regularization techniques (e.g., dropout, L1/L2 regularization), optimization algorithm, and batch size are common secondary parameters.	Learning rate, regularization techniques (e.g., dropout, L1/L2 regularization), optimization algorithm, and batch size are common secondary parameters that can be adjusted for optimal training.

2.10 Databases

The multimodal biometrics dataset SDUMLA-HMT was produced by a team of machine learning and applications researchers at Shandong University [43]. A variety of biometric information, including the face, iris, finger vein, fingerprint, and gait, was collected from 106 individuals and stored by SDUMLA-HMT. Images for 61 boys and 45 girls between the ages of 17 and 31 can be found in SDUMLA-HMT.

The dataset includes various pictures for each subject's five biometric characteristics. From the 106 subjects, 1060 iris pictures were collected by SDUMLA-HMT. An intelligent iris capture tool created by the University of Science and Technology of China captured 1,060 iris photos. The eye and the equipment were within a 6 cm to 32 cm range during the capture operation. Regarding facial photographs, SDUMLA-HMT has 8904 distinct pictures of 106 persons in various stances, with various facial emotions, illuminations, and accessories.

Lastly, SDUMLA-HMT contains 25,440 fingerprint images that were collected by a tool created by Wuhan University's Joint Lab for Intelligent Computing and Intelligent Systems. Different images of each subject's thumb, index, and middle fingers on both hands were taken during the capturing procedure.

Table 2.3 Details of SDUMLA-HMT datasets

traits	capture device	varieties	no of images	details of images	size of data	no of samples
face	7 ordinary digital cameras	poses (look upward, forward, and downward), facial expressions(smile, frown, surprise, and close eyes) accessories (glasses and hat) illuminations (3 types)	8,904	“bmp” files ,640×480 pixels	8.8GB	7× (3+4+2+3)×106
iris	intelligent iris capture device	5 images for each of the eyes	1,060	“bmp” format with 768 × 576 pixels in size	0.5GB	2×5×106
fingerprint	AES2501 swipe fingerprint scanner, FPR620 optical fingerprint scanner, FT-2BU capacitance fingerprint scanner, URU4000 optical fingerprint scanner, ZY202-B optical fingerprint scanner	thumb finger, index finger and middle finger of both hands	25,440	“bmp” format but the size varies according to the capturing sensors	2.2GB	6×5×8×106

2.11 Performance Measures

A multimodal biometric system's performance could be assessed using a variety of metrics. The following is an explanation of some of them:

Evaluation of a trained CNN model plays a crucial role in testing the model performance. A suitable evaluation metric is an essential key for ruling out poor models and obtaining the optimal one. In supervised learning, the Confusion Matrix is a commonly used evaluation method. The confusion matrix collects and organizes the predicted and ground truth labels in a matrix [44]. The number of samples correctly predicted by the classifier exists on the diagonal. These are divided into two categories; True Positives (TP) and True Negatives (TN).The confusion matrix can be utilized to calculate overall accuracy, precision, recall, specificity, and Sensitivity for each class. Fig 2.18 illustrate a two-class binary classification problem.

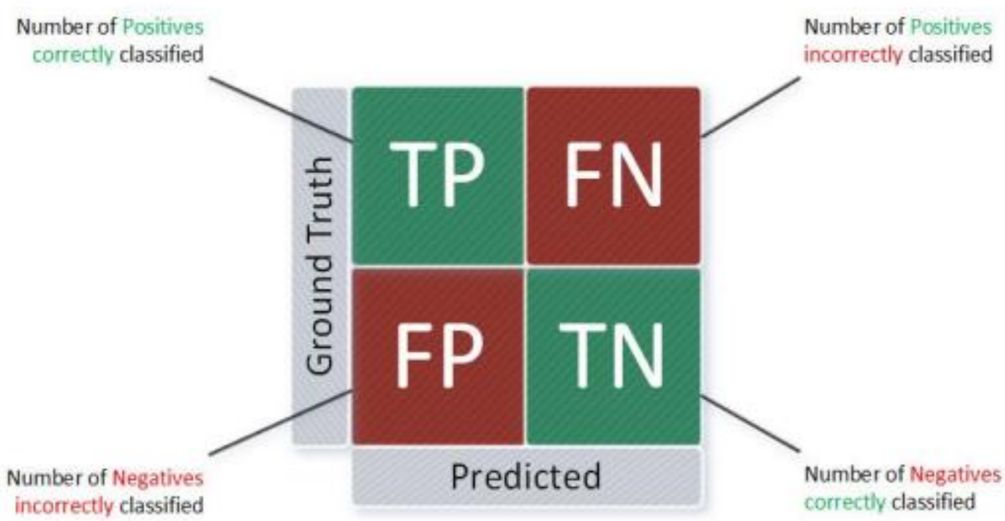


Figure 2.18 Confusion Matrix

2.11.1 Accuracy

It is an all-purpose metric that is employed in verification experiments. Accuracy is measured as a proportion of samples that have been successfully matched to people using equation.(2.15) [45].

Generally, the accuracy measures the ratio of correctly classified samples to the total number of samples.

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \qquad (2\text{-}15)$$

Chapter Three
the Proposed Model

3.1 Introduction

There are two types of biometric models: unimodal and multimodal biometric models, the latter of which is preferable. Multimodal models can be created by combining two or more unimodal systems.

In this proposed model, the feature level and score level fusion are projected to produce better outcomes than existing fusion techniques. The multi-modal biometric identification model that is suggested in this thesis fuses face, iris, and fingerprint modalities at the feature and score levels. This thesis also suggested three more single-modal biometric models in five different CNN models for comparing their outcomes to those of the multi-modal models. The SDUMLA-HMT database is used to test all three models.

3.2 Proposed Method

The aim of this thesis comprises five models as following:

1- Implemented single modal based face recognition model, which is employs just the face modality to personal identification, detailed in Section (3.3).

2- Implemented single modal based iris recognition model, which is employs just the iris modality to personal identification, detailed in Section (3.3).

3- Implemented single modal based fingerprint recognition model, which is employs just the fingerprint modality to personal identification, detailed in Section (3.3).

4- Proposed multimodal biometric identification model, which is fusion face, iris, and fingerprint modalities for personal identification at the feature and score levels.

Different CNNs networks are tested, to compare them and demonstrate the superiority of multimodal biometric models over single-modal models under fixed conditions. Each CNN network type is composed of the stages of preprocessing and proposed CNN structure.

3.3 The Proposed Single-modal Identification Model (Face, Iris and Fingerprint)

Single-modal personal identification system, consists of a face Identification model, an iris Identification model, and a fingerprint Identification model, is proposed in this thesis to address comparative and evaluation issues and to demonstrate the effectiveness of the proposed Multi-modal biometric identification model .

Prepossessing stage and the suggested single-modal CNN structure stage are the two primary stages of the single-module model, which is similar to the proposed Multi-modal model. In our research, we applying different types of CNN models as a pre-trained models to identify iris, face, and finger print such as (VGG16, ReseNet50 , MobilevNet , DenseNet , GoogleNet). The steps of proposed fingerprint Identification, iris Identification and face Identification system are shown in Fig 3.1, Fig 3.2 and Fig 3.3.

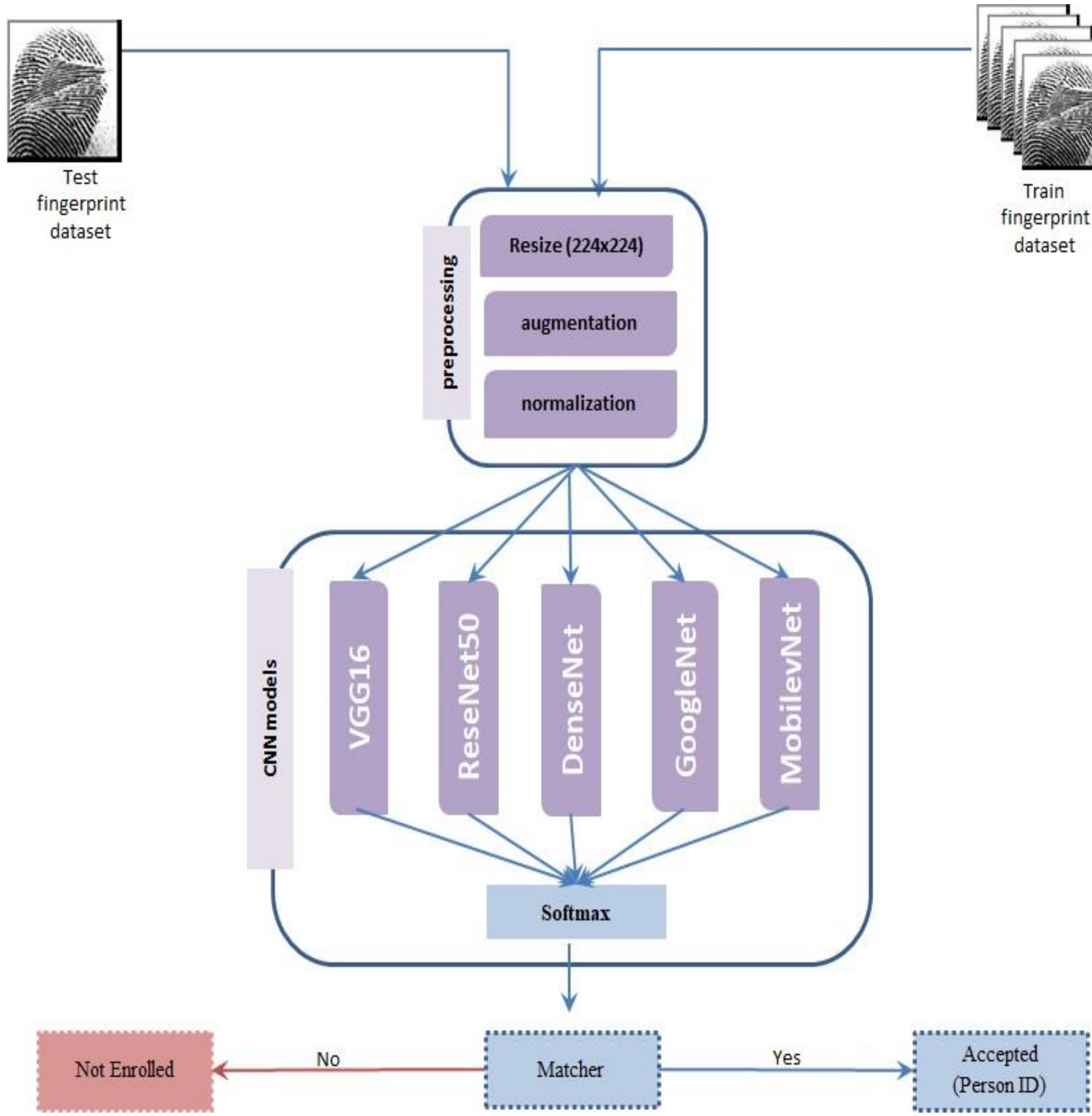


Figure 0.1 Block Diagram of the Single-modal for Fingerprint Identification System

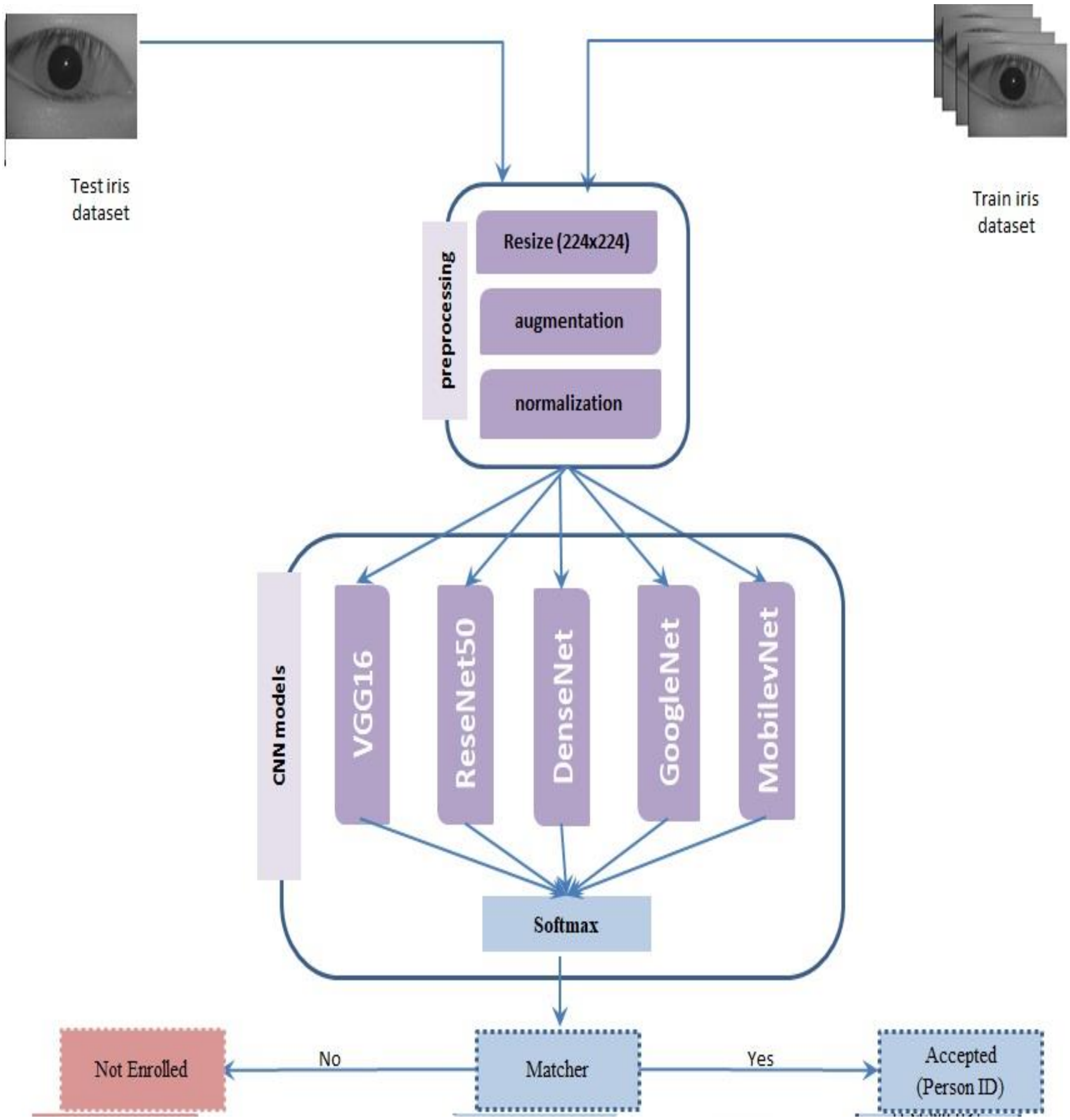


Figure 0.1 Block Diagram of the Single-modal for Iris Identification System

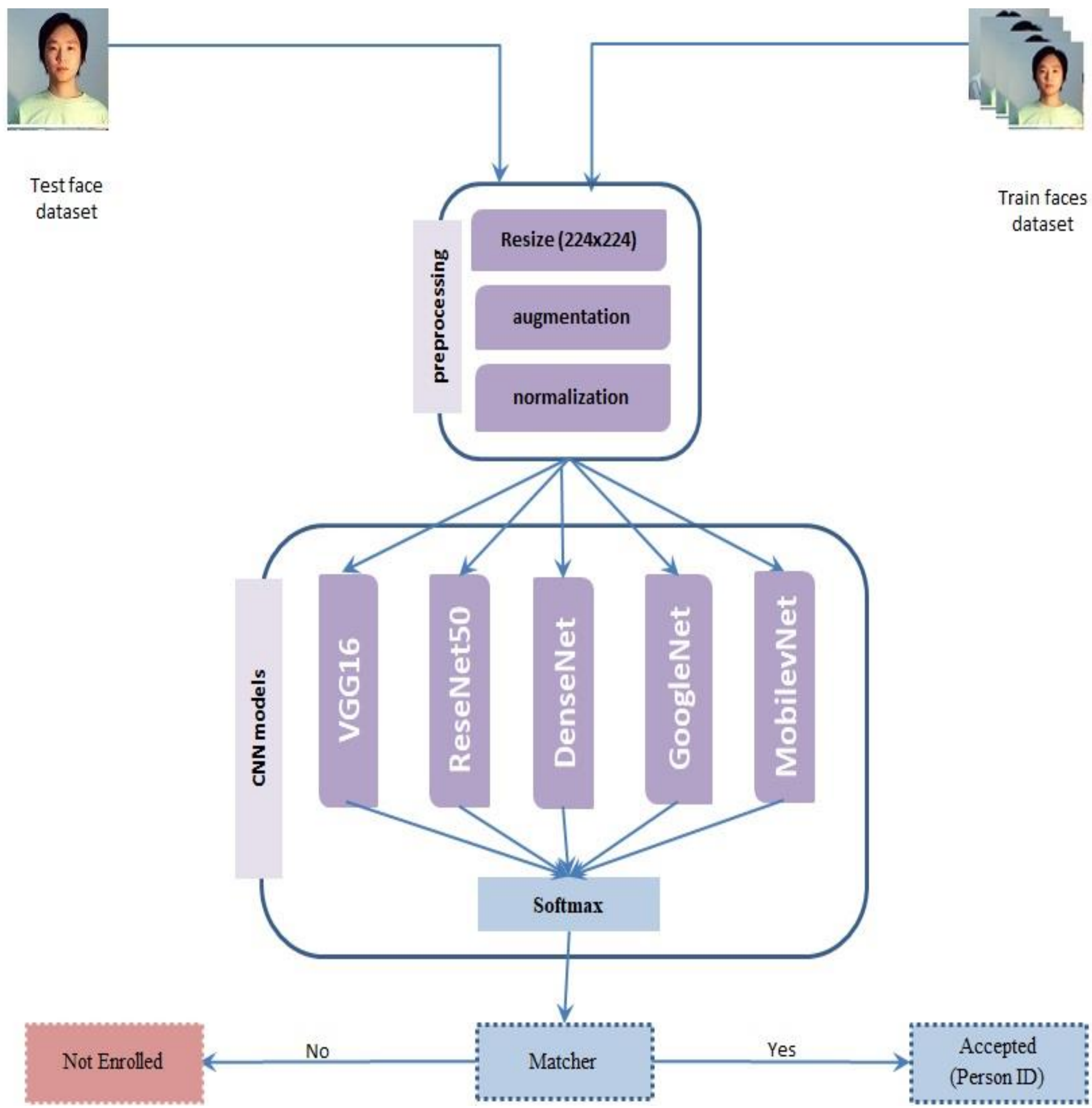


Figure 0.3 Block Diagram of the Single-modal for Face Identification System

3.3.1 Preprocessing Stage

The preprocessing methods used were image resizing, data augmentation and normalization. Images were downsized to 224 ×224 pixels and data augmentation was used to boost training data and decrease overfitting issues in order to make the images suitable for usage with the CNN model and data normalization to standardize the range and distribution of input data.

3.3.2 The Proposed Single-Modal CNN Structure stage

Algorithm (3.1) illustrates the process of one image feature extraction using the proposed CNN structure.

Algorithm (3.1): Feature Extraction	
goal	Extract three feature vectors from each face ,iris and fingerprint images
input	***xf*** // one face image, color image (224x224) pixel ***xi*** // one iris image, color image (224x224) pixel ***xp*** // one fingerprint image, color image (224x224) pixel ***K*** // initialized set of kernel weights for each convolutional layer ***W*** // initialized set of weights for the last layer FC1 ***N*** // dimension of feature-map, for first layer = 224
output	***FC1F*** // face feature vector of 1024 neuron ***FC1I*** // iris feature vector of 1024 neuron ***FC1P*** // fingerprint feature vector of 1024 neuron

Step1:

$\boldsymbol{XF^0} = \mathbf{xf}$// the input image will pass as input feature-map, where layer **L= 0**

$\boldsymbol{XP^0} = \mathbf{xp}$

$\boldsymbol{XI^0} = \mathbf{xi}$

$\boldsymbol{F^1 = 64}$ // the number of feature-maps in first layer, the number used in this thesis is 64

Step2: For **L = 1 to 5** // where L is the layer number

For **q = 0 to** $\boldsymbol{F^L}$// where $\boldsymbol{F^L}$ is number of feature-maps in **L**

For **i = 0 to N** // where **N = 64** is dimension of q feature-map

For **j = 0 to N**

Pass$\boldsymbol{XF^{L-1}}$ and K^Linto equation (2.10) for compute $\boldsymbol{zf^L_{q.i.j}}$

Pass $\boldsymbol{zf^L_{q.i.j}}$into equation (2.5) for compute $\boldsymbol{yf^L_{q.i.j}}$// Convolution L in Face-CNN

Pass $\boldsymbol{XP^{L-1}}$ and $\boldsymbol{K^L}$ into equation (2.10) for compute $\boldsymbol{zi^L_{q.i.j}}$

Pass $\boldsymbol{zi^L_{q.i.j}}$ into equation (2.5) for compute $\boldsymbol{yi^L_{q.i.j}}$// Convolution L in Iris-CNN

Pass $\boldsymbol{XP^{L-1}}$ and $\boldsymbol{K^L}$ into equation (2.10) for compute $\boldsymbol{zp^L_{q.i.j}}$

Pass $\boldsymbol{zi^L_{q.i.j}}$ into equation (2.5) for compute $\boldsymbol{yp^L_{q.i.j}}$// Convolution L in Fingerprint-CNN

u = int(i/2) // indexes for max-pooling feature-map

v = int(i/2)

Pass $\boldsymbol{yf}^{\boldsymbol{L}}_{\mathbf{q}.(\mathbf{i}+\mathbf{a}).(\mathbf{i}+\mathbf{b})}$ to equation (2.12) for compute $\boldsymbol{xf}^{\boldsymbol{L}}_{\mathbf{q}.\mathbf{u}.\mathbf{v}}$ where **a, b ∈(0,1)** // Max pooling **L** in face-CNN

Pass $\boldsymbol{yi}^{\boldsymbol{L}}_{\mathbf{q}.(\mathbf{i}+\mathbf{a}).(\mathbf{i}+\mathbf{b})}$ to equation (2.12) for compute $\boldsymbol{xi}^{\boldsymbol{L}}_{\mathbf{q}.\mathbf{u}.\mathbf{v}}$,where **a, b ∈(0,1)** // Max pooling **L** in Iris-CNN

Pass $\boldsymbol{yp}^{\boldsymbol{L}}_{\mathbf{q}.(\mathbf{i}+\mathbf{a}).(\mathbf{i}+\mathbf{b})}$to equation (2.12) for compute $\boldsymbol{xp}^{\boldsymbol{L}}_{\mathbf{q}.\mathbf{u}.\mathbf{v}}$,where **a, b ∈(0,1)** // Max pooling L in Fingerprint -CNN

Next **j**

Next **i**

Next **q**

$\boldsymbol{F}^{\boldsymbol{L}+\mathbf{1}} = \boldsymbol{F}^{\boldsymbol{L}} \times \mathbf{2}$ // duplicate the number of feature-maps according to structure

N = N / 2 // reduce the dimensions parameter to be equal to dimensions of pooled feature-map

Next **L**

Step3: Pass **FLF** into equation (2. 9) for compute **DF**

Pass **FLI** into equation (2. 9) for compute **DI**

Pass **FLP** into equation (2. 9) for compute **DP**

Step4: Pass **ZF** into equation (2.5) for compute **FC1F**

Pass **ZI** into equation (2.5) for compute **FC1I**

Pass **ZP** into equation (2.5) for compute **FC1P** // when **FC1** is a vector of **1024** values

End

Algorithms (3.2) and (3.2) illustrated the processes of feature fusion steps by applying two approaches (feature and score level).

Algorithm (3.2): Feature Fusion(feature level)	
goal	Concatenation of feature vectors of face ,iris and fingerprint , for compute a single vector of potentials and Loss value
input	***FCF*** // face feature vector, which output from algorithm (3.1) ***FCI*** // iris feature vector, which output from algorithm (3.1) ***FCP*** // fingerprint feature vector , which output from algorithm (3.1) ***Y*** // vector of **106** label , one for each person (Desired outputs) ***W*** // weights of **FC2** layer ***N*** // dimension of **FC**, which = **1024** here
output	**P** // potentials vector contains 106 value **Loss** // one value represent error in prediction of an individual

Step1: **For i = 0 to N** // where N is dimension of **FC**

$\boldsymbol{FC_i = FC_{Fi} + FC_{Ii} + FC_{Pi}}$ // feature fusion

Next i

Step2: Pass **FC** into equation (2. 9) for compute **D** // apply Dropout

Step3: Pass **D** and **W** into equation $\boldsymbol{z = W^T \mathbf{x} + \mathbf{b}}$ for compute **P** // **P** is the Potentials vector // contain **106** value

Step4: Pass **P** into equation (2.5) for compute **FC2**

Step5: Pass **FC2** into equation (2.6) for compute **S** // probabilities vector computed // using **SoftMax** function

Step6: Pass **S** and **Y** into equation (2.11) for computing **Loss** //one value represents //the error of the neural network prediction

End

Algorithm (3.3): Feature Fusion(score level)	
goal	A sum rule of three feature vectors of face ,iris and fingerprint , for compute a single vector of potentials and Loss value
input	***FCF*** // face feature vector, which output from algorithm (3.1) ***FCI*** // iris feature vector, which output from algorithm (3.1) ***FCP*** // fingerprint feature vector , which output from algorithm (3.1) ***Y*** // vector of **106** label , one for each person (Desired outputs) ***W*** // weights of **FC2** layer ***N*** // dimension of **FC**, which = **1024** here
output	**P** // potentials vector contains 106 value **Loss** // one value represent error in prediction of an individual

Step1: Pass **FC** into equation (2. 9) for compute **D** // apply Dropout

Step2: Pass **D** and **W** into equation $\boldsymbol{z = W^T}\ \mathbf{x + b}$ for compute **P** // **P** is the Potentials vector // contain **106** value

Step3: Pass **P** into equation (2.5) for compute **FC2**

Step4: Pass **FC2** into equation (2.6) for compute **S** // probabilities vector computed // using **SoftMax** function

Step5:

For i = 0 to N // where N is dimension of **FC**

$\mathbf{FCi} = \sum_{I=1}^{n} \mathbf{WS}$ //Fusion_score = wf × score f + wi × score i + wp × score p

Next i

Step6: Pass **S** and **Y** into equation (2.11) for computing **Loss** //one value represents //the error of the neural network prediction

End

3.4 Proposed System for Multimodal Biometric Identification

This thesis proposes a multimodal biometric identification system for the fusion of face, iris, and fingerprint modalities at feature level and score level. The multimodal biometric system is more reliable and accurate than unimodal biometric system.

The system is built in three stages: first is preprocessing stage, which prepares the input images for the following stage, the second stage with the suggested CNN structure, which is utilized for feature extraction, feature fusion, and matching and third stage is fusion. The steps for each of these stages are shown in Fig (3.4) and (3.5)

The extraction and classification of features were performed using the CNN method.

The three traits face, iris, and fingerprint were fused utilizing feature-level fusion and score-level fusion. The SDUMLA-HMT dataset [43] was chosen to train the multimodal system, conduct preliminary tests, and assess its effectiveness.

At the feature and score levels, the three CNNs single models (face, iris, and fingerprint) were fused. The feature-level fusion was chosen since it is carried out before the matching module and is typically successful because the data includes rich details about the features. To significant trade-off between the ease of combining the data from the traits and improved information content, score level fusion was also chosen. Fig 3.4 illustrates the feature level fusion strategy used in this study's multimodal CNN model, whereas Fig 3.5 illustrates the score level fusion approach.

The Figures demonstrate how fingerprint, iris, and face images are first taken from the dataset. The images are then submitted to some preprocessing operations, such as image resizing. Three CNNs are then fused after each of the biometric attributes has been fed into its own CNN model. The features are merged prior to the softmax classifier in Fig 3.4 because the fusing is done at the feature level. In Fig 3.5 the scores are merged after the softmax classifier, the fusing is done at the score level. The user identity is the final output of the fused model. The following subsections provide descriptions of the model's various components.

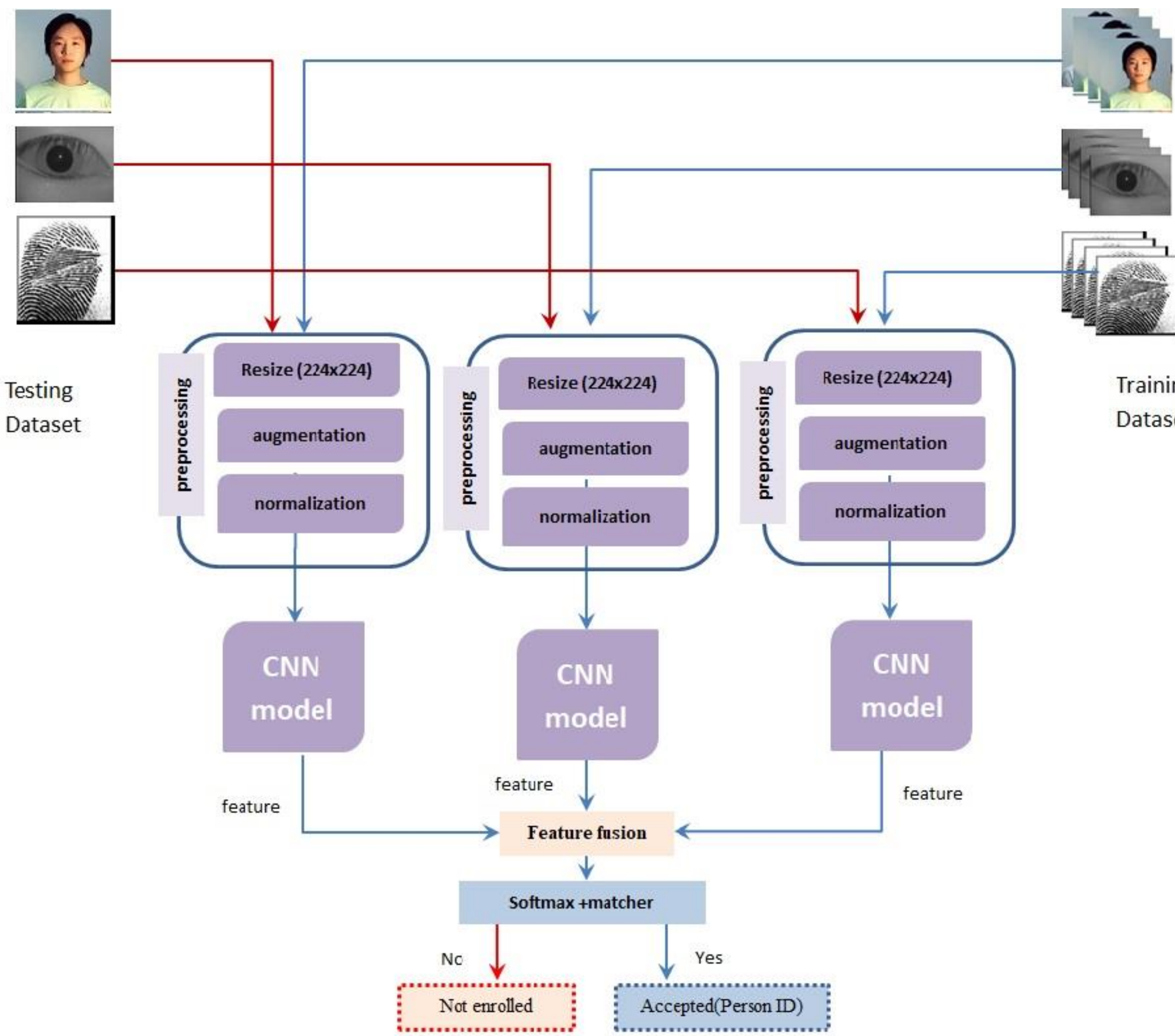


Figure 3.4 Structure of Multimodal Biometric Model Using Feature Level Fusion Approach

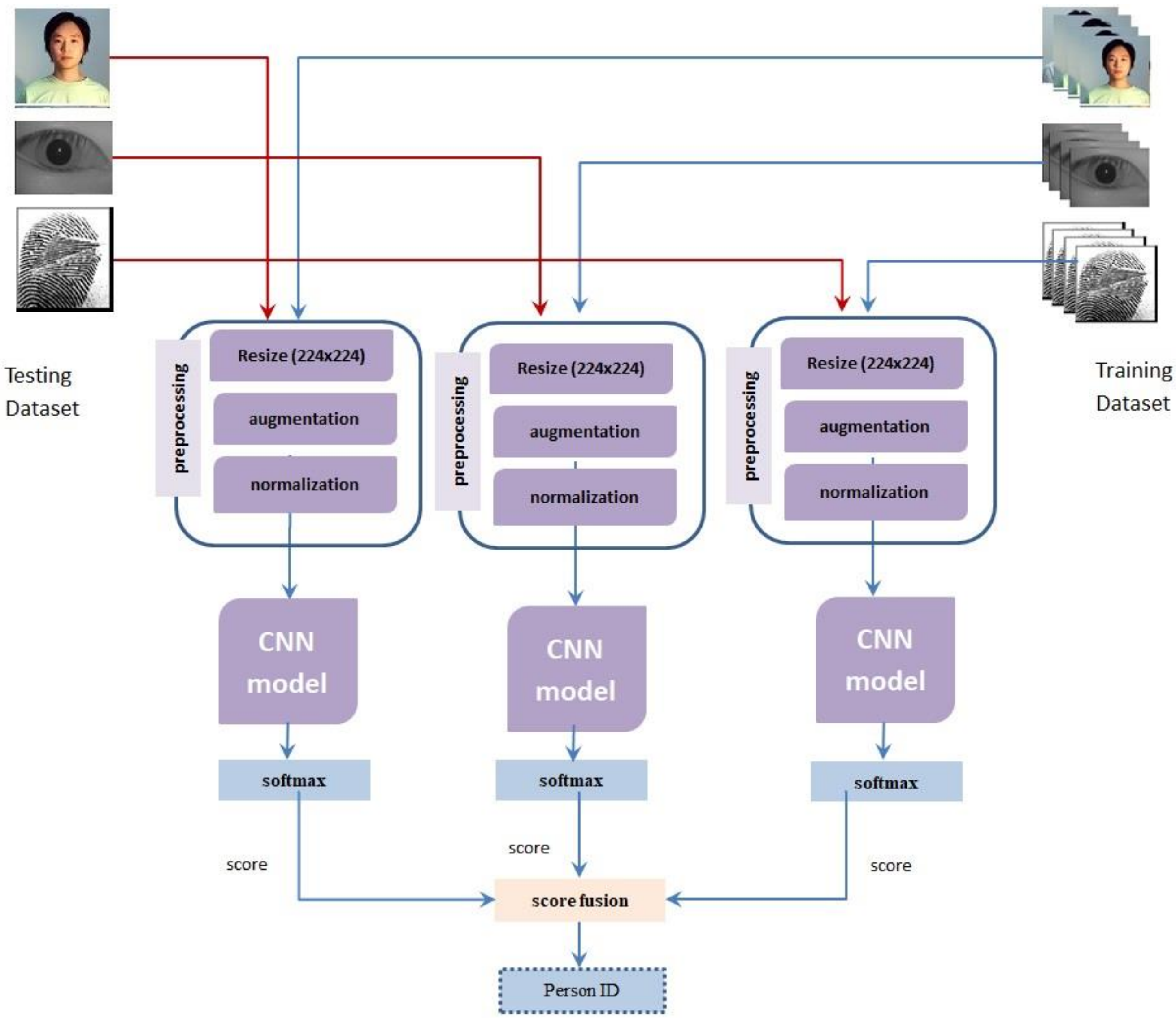


Figure 0.5 Structure of Multimodal Biometric Model Using Score-Level Fusion Approach

The proposed system needs to be trained on all images of enrolled persons to be able to identify them correctly in the matching Phase. The proposed system is trained using training database which described in chapter 2, Algorithms (3.4) illustrated the processes of proposed model steps by applying five type of CNN models (VGG16, ReseNet50 , MobilevNet , DenseNet , GoogleNet) which described in chapter 2 and two approaches of fusion (feature and score level).

Algorithm (3.4): proposed model	
goal	Training the model to produce a set of trained weights and kernels
input	**XF** // All face images in Train_Faces database **XI** // All iris images in Train_Iris database **XP** // All fingerprint images in Train_Fingerprint database **Y** // All person's labels (desired output) epochs // number of iterations needed for training, 100 epochs used
output	**W** // trained sets of weights **K** // trained sets of kernel weights
Step1: Preprocessing all image in XF ,XI and XP as described in the section (3.3.1) **Step2**: Initialize the set of kernels weights K for all convolutional layers Initialize the set of weights W for all fully connected layers **Step3**: **For epoch = 1 To epochs** **For i =1 to N** //where **N** is the count of images in a database // Forward phase Pass **XFi, XIi, XPi, K , W l** into algorithm (3.2) of feature extraction to produce **FCfi , FCpi , FCii**//one of 5 cnn model (VGG16, ReseNet50, MobilevNet, DenseNet, GoogleNet) Pass **FCfi ,FCpi , FCii, Yi , W l** into algorithm (3.3) of fusion(feature or score) **Next i** **Next epoch** **End**	

Chapter Four

Experimental Results and Evaluation

4.1 Introduction

In this chapter, the performance results of suggested recognition models in the previous chapter is recorded (Multimodal biometric identification model, Face identification model, iris identification model, and fingerprint identification model). Each model is composed of two phases, training and testing phases, and each phase is consisted of two stages, preprocessing stage and a proposed CNN structure stage. This chapter is covered three main topics, the databases and its arrangement and the evaluation of the proposed models.

4.2 Experimental Setup

The following are the hardware and software specifications which are used to implement the proposed recognition models.

A. Hardware Specifications:

Personal computers are used to implement the proposed models is dell Latitude E5570 Work-Station, with specifications as follow:

1-CPU: Intel(R) Core (TM) i7-6820HQ CPU @ 2.70GHz 2.70 GHz

2-RAM: 8 GB

3-GPU: NVIDIA Quadro K1000M, Parallel Processor Cores: 192 of 384 cores, dedicated Memory: 2 G

B. Software Specifications:

Python 3.6 is the programming language used to construct the proposed recognition models, which is installed on Windows 10.

Python and the Tensor Flow open-source framework are combined to construct the CNN code quickly. Tensor Flow is a Google open-source software library that focuses on working effectively with tensors (tensors are generalizations of vectors and matrices). Tensor Flow supports high speed for execution since Neural Networks implementation depends on matrix computations. The proposed model is created using the Keras Python library.

4.3 Databases Preparations

This study employed SDUMLA-HMT datasets [43]. As the first stage of constructing the suggested multimodal system, training and testing the (face, iris and fingerprint) unimodal identification models. After that, SDUMLA-HMT is employed to assess how well the suggested multimodal system performs.

The SDUMLA-HMT dataset including the face, iris and fingerprint, was collected from 106 individuals images for 61 men and 45 women between the ages of 17 and 31.

All the biometric traits with the same person id are captured from the same subject. All details illustrates in table (4.1)

Table 4.1 Details of SDUMLA-HMT datasets

traits	**capture device**	**varieties**	**no of images**	**details of images**	**size of data**	**no of samples**
face	7 ordinary digital cameras	poses (look upward, forward, and downward), facial expressions(smile, frown, surprise, and close eyes) accessories (glasses and hat) illuminations (3 types)	8,904	“bmp” files ,640×480 pixels	8.8GB	7× (3+4+2+3)×106
iris	intelligent iris capture device	5 images for each of the eyes	1,060	“bmp” format with 768 × 576 pixels in size	0.5GB	2×5×106
fingerprint	AES2501 swipe fingerprint scanner, FPR620 optical fingerprint scanner, FT-2BU capacitance fingerprint scanner, URU4000 optical fingerprint scanner, ZY202-B optical fingerprint scanner	thumb finger, index finger and middle finger of both hands	25,440	“bmp” format but the size varies according to the capturing sensors	2.2GB	6×5×8×106

Using percentages (60:20:20), the images of each topic (class) in SDUMLA-HMT were split into training, validation, and testing sets, with 60% used for training, 20% for validation, and 20% for testing. For training, validation, and testing, the dataset images were divided into three folders, each of which comprises samples for each subject. The validation set was used to assess the final model fit using only the forward pass, while the training set was utilized to train and fit the deep learning model utilizing continuous forward and backward passes through it. The accuracy metric, which can be used to evaluate system performance, was the main emphasis of the system evaluation procedure. Accuracy can be used to assess the suggested models and investigate the impact of the various hyperparameters. It can be measured as the percentage of images that were correctly categorized to all of the images.

4.3 Experimental Results and Discussion

Tables (4.2) and (4.3) record the performance results of the unimodal and multimodal models, respectively. These tables describe the testing time and identification accuracy, precision and recall findings from the experiments that were conducted. The findings show that, in compared to unimodal models, the multimodal biometric model achieved greater accuracy rates. This demonstrates that multimodal biometrics offers a highly effective technique to increase the accuracy rates of a biometric model, as initially intended.

Table 4.2 Accuracy Results Using Unimodal Biometrics Models

Biometric Model (traits)	accuracy of each CNN model				
	VGG16	ReseNet50	MobilevNet	DenseNet	GoogleNet
face	99.94%	99.96%	84.53%	80.04%	62.73%
iris	99.87%	99.54%	87.64%	80.07%	62.65%
fingerprint	98.21%	97.59%	81.63%	58.13%	57.19%

Table 4.3 Accuracy Results Using the Multimodal Biometrics Models

Fusion Approach	accuracy of each CNN model				
	VGG16	ReseNet50	MobilevNet	DenseNet	GoogleNet
Feature level fusion (face, iris and fingerprint)	97.55%	96.15%	86.23%	72.41%	59.90%
Score level fusion (face, iris and fingerprint)	99.37%	98.05%	89.17%	75.34%	61.80%

Depending on the kind of fusion approach have been used, a comparison between the proposed multimodal model's results and those from a unimodal model was made, as shown in Table 4.2 When using VGG16, it is important to note that the suggested multimodal model with Feature level fusion (accuracy of 97.55%) outperformed the multimodal model with Score level fusion (accuracy of 99.37%). In comparison to performance based on existing CNN models, a better recognition accuracy was obtained by combining the three qualities at the score level by utilizing

VGG16 and ReseNet50. The score level fusion, it should be highlighted, had a higher detection performance than the feature level fusion. This is connected to the softmax classifier; in the feature level fusion, the softmax classifier is given a vector containing a combination of different feature sets retrieved from the various biometric attributes, and the classifier then generates the final score. Three softmax classifiers were used in the score level fusion, and each one was given a vector of one trait's features to create a score. These scores were then fused using the fixed rule technique.

A comparison of the proposed method's performance multimodal and single modal based on the feature fusion method and score fusion method are shown in Fig 4.1 and Fig 4.2 These two figures show the proposed multimodal fusion biometric recognition model has achieved a better accuracy in score level than feature level. The performance of the proposed unimodal model with all CNN types that used (VGG16 , ReseNet50 , MobilevNet , DenseNet , GoogleNet) are also illustrated.

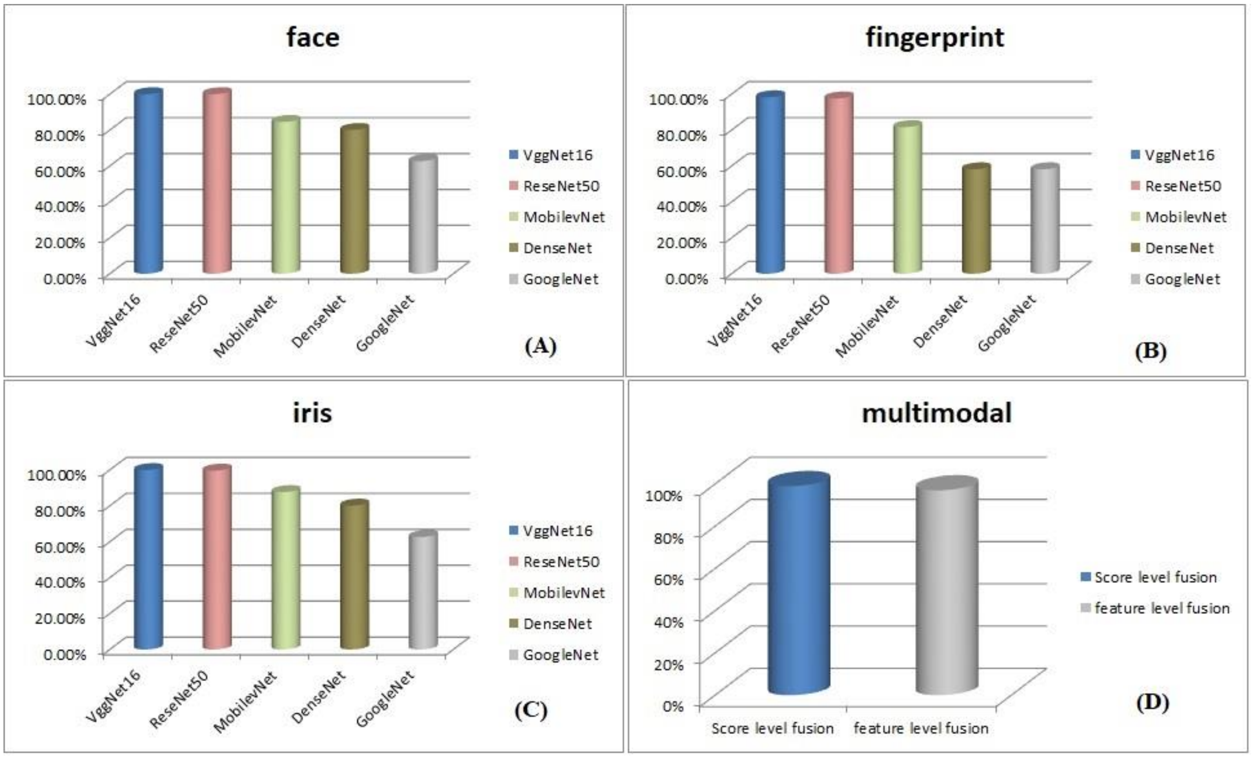
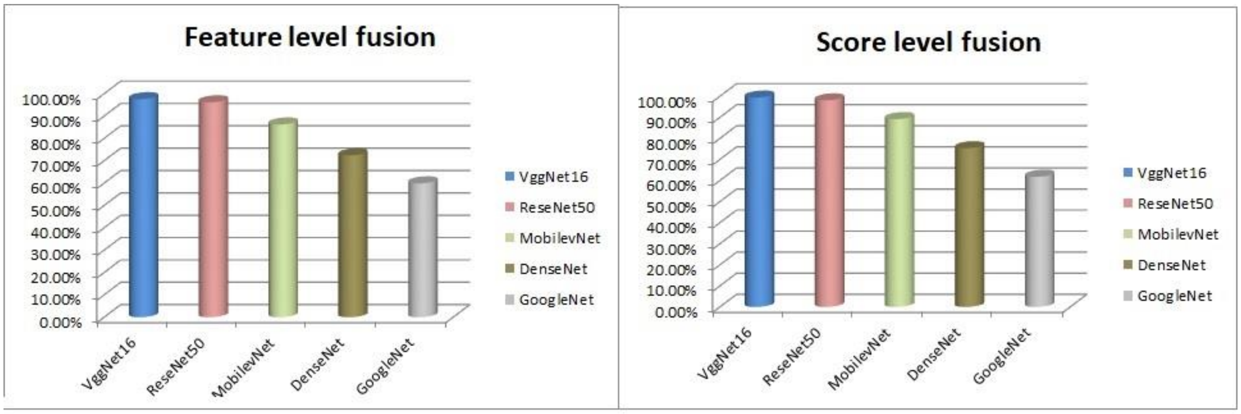


Figure 4.1 (A), (B) and (C) Presented Accuracy of Proposed Single Modal with all CNN types (VGG16 , ReseNet50 , MobilevNet , DenseNet , GoogleNet) ,(D) Presented Accuracy of Proposed Multimodal with VGG16

The model with score fusion has achieved better results than model with feature fusion. Accuracy greater than 97.55% has been achieved by model with feature fusion, while model with score fusion achieved identification accuracy of 99.37%.



Figur4.2 Accuracy of Proposed Framework (Multimodal)

Score level fusion, is reasonably simple to use, has less noise, and has the most information for all input data. In addition, compared to feature-level fusion, it has the most comprehensive data about the input pattern.

Chapter Five

Conclusion and Future Works

5.1 Conclusion

In this thesis, effective and efficient (face, iris and fingerprint) multimodal biometric model has been developed. The proposed model employed the CNN deep learning algorithm to identify the user. To our knowledge, this is the first study to investigate the use of deep learning algorithms for a multimodal biometric with these three traits. Moreover, as mentioned earlier, no work has been conducted on the multimodal identification biometric model by applying several types of CNN models such as (VGG16 , ReseNet50 , MobilevNet , DenseNet , GoogleNet).

The combination of three biometrics produces a robust multimodal biometric model. Score level fusion and feature level fusion are adopted to perform the fusion of face, iris and fingerprint features. The experiment is performed on the SDUMLA-HMT database. Generally, the score level fusion approach obtained better accuracy (99.37%) than the feature level fusion (97.55%) in biometric models due to several reasons:

1. Independence of Biometric Modalities: Score-level fusion treats different biometric modalities as independent sources of information. It combines the matching scores or similarity measures obtained from each modality, regardless of the specific features used to compute those scores. This independence allows for the combination of modalities with different feature extraction methods, matching algorithms, or even different levels of information quality.

2. Robustness to Feature Variability: Feature-level fusion requires the alignment and normalization of features extracted from different modalities to a common representation.

3. Handling of Unequal Feature Contributions: In feature-level fusion, the features from different modalities are combined before the matching process. However, not all features contribute equally to the final decision.

4. Reduced Complexity: Score-level fusion tends to be simpler and computationally more efficient compared to feature-level fusion. It directly operates on the matching scores, which are typically scalar values, rather than complex multi-dimensional feature vectors.

However, it's important to note that the choice between score-level and feature-level fusion depends on the specific characteristics of the biometric model, the nature of the biometric modalities, and the requirements of the application.

5.2 Future Works

Some potential areas for future work could include:

1- Fusion Architectures: Explore and develop new fusion architectures specifically designed for deep learning-based biometric fusion. This could involve investigating novel neural network architectures, such as multi-scale or attention-based fusion networks, that can effectively combine information from different biometric modalities or fusion levels.

2- Explainability and Interpretable Fusion: Deep learning models are often considered black boxes, making it challenging to interpret their decisions. Future work can focus on developing techniques to provide interpretable explanations for the fusion process in deep learning-based biometric models.

3- Privacy-Preserving Deep Fusion: Explore privacy-preserving techniques for deep learning-based biometric fusion. Investigate methods to perform fusion while preserving the privacy of biometric data, such as federated learning, secure aggregation, or differential privacy. Consider the trade-off between privacy and performance to design efficient and effective privacy-preserving fusion approaches.

4- Robustness to Adversarial Attacks: Deep learning models are vulnerable to adversarial attacks, including attacks on biometric models. Future work can focus on enhancing the robustness of deep learning-based biometric fusion against adversarial attacks.

5- Investigate techniques to optimize the computational requirements, memory usage, and energy consumption of the fusion models. Explore hardware acceleration, parallel computing, or model compression techniques to enable real-time and resource-efficient fusion.

References

References

[1] B. Arjun and H. N. Prakash, 2021 .Multimodal Biometric Recognition System Using Face and Finger Vein Biometric Traits with Feature and Decision Level Fusion Techniques.pdf.

[2] Singh, M., Singh, R., & Ross, A. (2019). A comprehensive overview of biometric fusion. *Information Fusion*, *52*, 187-205.

[3] Dorizzi, B. (2012). Introduction to Biometrics. *Signal and Image Processing for Biometrics*, 1-13.

[4] Ammour, B., Boubchir, L., Bouden, T., & Ramdani, M. (2020). Face–iris multimodal biometric identification system. *Electronics*, *9*(1), 85.

[5] Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146-157.

[6] Nagpal, C., & Dubey, S. R. (2019, July). A performance evaluation of convolutional neural networks for face anti spoofing. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

[7] Paliwal, M., & Kumar, U. A. (2009). Neural networks and statistical techniques: A review of applications. *Expert systems with applications*, *36*(1), 2-17.

[8] Alay, N., & Al-Baity, H. H. (2020). Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits. *Sensors*, *20*(19), 5523.

[9] Mahmoud, R. O., Selim, M. M., & Muhi, O. A. (2020). Fusion time reduction of a feature level based multimodal biometric authentication system. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, *12*(1), 67-83.

[10] Yadav, A. K. (2021). Deep learning approach for multimodal biometric recognition system based on fusion of iris, fingerprint and hand written signature traits. *Turkish Journal of Computer and Mathematics Education*, *12*(11), 1627-1640.

[11] Kamlaskar, C., & Abhyankar, A. (2021). Iris-fingerprint multimodal biometric system based on optimal feature level fusion model. *AIMS Electronics and Electrical Engineering*, *5*(4), 229-250.

[12] Vijayakumar, T. (2021). Synthesis of palm print in feature fusion techniques for multimodal biometric recognition system online signature. *Journal of Innovative Image Processing (JIIP)*, *3*(02), 131-143.

[13] Channegowda, A. B., & Prakash, H. N. (2021). Multimodal biometrics of fingerprint and signature recognition using multi-level feature fusion and deep learning techniques. *Indones. J. Electr. Eng. Comput. Sci*, *22*(1), 187.

[14] Sarangi, P. P., Nayak, D. R., Panda, M., & Majhi, B. (2022). A feature-level fusion based improved multimodal biometric recognition system using ear and profile face. *Journal of Ambient Intelligence and Humanized Computing*, 1-32.

[15] Naamha, E. Q., & Rahma, A. M. (2017). Fingerprint Identification and Verification System Based on Extraction of Unique ID. *PhD diss., University Of Technology*.

[16] Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, *14*(1), 4-20

[17] Razoqi, Z. N. (2017). *Palm Vein Recognition Using Centerline Extraction* (Doctoral dissertation, University of Technology).

[18] Preethi, M., Vaidya, D., Kar, S., Sapkal, A. M., & Joshi, M. A. (2015). Person authentication using face and palm vein: a survey of recognition and fusion techniques. *Int. J. Technol. Enhanc. Emerg. Eng. Res.*, *3*(03), 55-69.

[19] El Emary, I. M., & Abdulkareem, M. M. (2011, July). Fingerprints registration using genetic algorithm. In *Proceedings of the 15th WSEAS international conference on Computers* (pp. 91-97).

[20] Alay, N., & Al-Baity, H. H. (2020). Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits. *Sensors*, *20*(19), 5523.

[21] van Veen, F. (2016). The neural network ZOO. The Asimov Institute Blog posted on September 14.

[22] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*, 1-74.

[23] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27-48.

[24] Shaheen, F., Verma, B., & Asafuddoula, M. (2016, November). Impact of automatic feature extraction in deep learning architecture. In *2016 International conference on digital image computing: techniques and applications (DICTA)* (pp. 1-8). IEEE.

[25] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

[26] Serra, X., & Castán, J. (2017). Face recognition using Deep Learning. *Catalonia: Polytechnic University of Catalonia*, *78*.

[27] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

[28] Calli, E. (2017). Faster Convolutional Neural Networks.

[29] Mouton, C., Myburgh, J. C., & Davel, M. H. (2020, December). Stride and translation invariance in CNNs. In *Southern African Conference for Artificial Intelligence Research* (pp. 267-281). Cham: Springer International Publishing.

[30] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing.

[31] Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. *Genetic programming and evolvable machines*, *19*(1-2), 305-307.

[32] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

[33] Parsania, P., & Virparia, P. V. (2014). A review: Image interpolation techniques for image scaling. *International Journal of Innovative Research in Computer and Communication Engineering*, *2*(12), 7409-7414.

[34] Acharya, T., & Tsai, P. S. (2007). Computational foundations of image interpolation algorithms. *Ubiquity*, *2007*(October), 4-1.

[35] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

[36] Obaid, H. S., Dheyab, S. A., & Sabry, S. S. (2019, March). The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. In *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)* (pp. 279-283). IEEE.

[37] Purohit, H., & Ajmera, P. K. (2021). Optimal feature level fusion for secured human authentication in multimodal biometric system. *Machine Vision and Applications*, *32*, 1-12.

[38] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[39] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[40] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[41] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[42] Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L., & Weinberger, K. Q. (2019). Convolutional networks with dense connectivity. *IEEE transactions on pattern analysis and machine intelligence*, *44*(12), 8704-8716.

[43] Yin, Y., Liu, L., & Sun, X. (2011). SDUMLA-HMT: A multimodal biometric database. In *Biometric Recognition: 6th Chinese Conference, CCBR 2011, Beijing, China, December 3-4, 2011. Proceedings 6* (pp. 260-268). Springer Berlin Heidelberg.

[44] Sanchez, S. A., Romero, H. J., & Morales, A. D. (2020, May). A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. In *IOP Conference Series: Materials Science and Engineering* (Vol. 844, No. 1, p. 012024). IOP Publishing.

[45] Radzi, S. A., Hani, M. K., & Bakhteri, R. (2016). Finger-vein biometric identification using convolutional neural network. *Turkish Journal of Electrical Engineering and Computer Sciences*, *24*(3), 1863-1878.

دمج الصفات البايومترية المتعددة للانسان بأستخدام تقنيات التعلم العميق

الخلاصة

نظرًا لتزايد هجمات الانتحال بشكل كبير، يتم إجراء الكثير من الأبحاث حول أنظمة الأمان البيومترية. يهتم الباحثون أكثر بالقياسات الحيوية متعددة الوسائط لتوفير أمان أفضل باستخدام تطبيقات القياسات الحيوية.

دمج القياسات الحيوية البشرية هو نهج يجمع بين طرائق القياسات الحيوية المتعددة لتعزيز دقة المصادقة وموثوقية أنظمة تحديد الهوية البشرية. تم تصميم نظام التعلم العميق للتعرف على الأشخاص باستخدام القياسات الحيوية للوجه وقزحية العين وبصمات الأصابع.

تقوم الشبكات العصبية التلافيفية (CNNs) التي تشكل بنية النظام باستخراج الميزات من الصور واستخدام مصنف Softmax لتصنيفها.

تم اختبار العديد من أساليب الدمج لدمج نماذج CNN لفحص تأثيرها على أداء التعرف. ونتيجة لذلك، تم استخدام نهج الدمج على مستوى النقاط والميزات. بالإضافة إلى ذلك، يتم تنفيذ خوارزميات CNN (VGG16، وResеNet50، وMobilevNet، وDenseNet، وGoogleNet) لإنشاء أنظمة بيومترية أحادية الوجه وقزحية العين وبصمات الأصابع لمقارنة نتائجها وإثبات كفاءة وتفوق النظام متعدد الوسائط.

يستخدم التدريب النموذجي والاختبار بالإضافة إلى تقييم النموذج مجموعة البيانات البيومترية الفعلية متعددة الوسائط SDUMLA-HMT التي يمكن الوصول إليها بشكل عام. باستخدام تقنية الدمج على مستوى الميزات، بلغت دقة النموذج المقترح ٩٧,٥٥%، ومع نهج الدمج على مستوى النتيجة، بلغت ٩٩,٣٧%، مما يتفوق بسهولة على الأساليب الحديثة الحالية، وفقًا للنتائج.

وبالتالي، فإن نموذج التعريف البيومتري المتعدد المقترح يمكن أن يمثل تقنية ترخيص قوية.

Imam Ja'afar Al-Sadiq University

IEEE

AICCIT 2023

3rd Al-Sadiq International Conference on Communication and Information Technology

https://sadiquni.com/conferences/public/index.php/index

Date:14/5/2023

LETTER OF ACCEPTANCE

Dear authors: *Hawraa A. Hussain, Hawraa H. Abbas*

We are pleased to inform you that, your manuscript "***Biometric Fusion Approaches Based on Deep Convolutional Neural Network***" has been accepted in the Third Al-Sadiq International Conference on Communication and Information Technology (3^{rd} AICCIT-2023), and it will be indexing in Scopus. The final decision of publication in IEEE Xplore is subject terms and conditions of Conference Scientific Committee and IEEE.

Please do not hesitate to contact the secretariat of AICCIT-2023 by sending email to: aiccit23@sadiq.edu.iq

Yours Sincerely,
The Program Committee of AICCIT-2023

Dr. Ali Hashim Abbas
Chairman of the 3rd AICCIT 2023

Prog. Dr. Sattar B. Sadkhan
Representative of IEEE \ Iraq

Kerbala Journal for Engineering Science

https://kjes.uokerbala.edu.iq/

A Survey on Multi-biometric Fusion Approaches

Hawraa abed Al-Kareem Hussain *, Hawraa Hassan Abbas **

*Department of Computer, College of Science for Women, University of Babylon, Babylon, Iraq.

E-mail: Haw.prog@gmail.com,

** Department of Electrical & Electronic Engineering, College of Engineering, University of Kerbala, Kerbala, Iraq.

E-mail: hawraa.hussain.gsci5@student.uobabylon.edu.iq

Received: 24 November 2022; Revised: 19 June 2023; Accepted: 28 June 2023

ABSTRACT

The goal of biometrics is to reliably and robustly identify people based on their unique personal characteristics, primarily for security and authentication needs, but also to identify and track the users of more intelligent applications. Fingerprints, iris, palm prints, faces, and voices are common biometric modalities, but there are countless additional biometrics, such as stride, ear image, retina, DNA, and even behaviour. An automatic way to identify a person depends on just one (single-modal biometrics) or a mix of (multi-modal biometrics). A fusion of two or more images can create multi-modal biometrics, and the resulting fused image will be more secure. Various fusion methods are now available and may be categorised by the degree of information they combine. This paper discusses different fusion approaches implemented in multi-modal biometrics to identify human biometrics by extracting features and classifying images. It also describes the datasets that were used and the results and conclusions that were obtained.

Keywords:

Biometric Fusion; Deep Learning; Convolutional Neural Networks; Recognition System; Multi-Modal Biometric System; Level Fusion.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل-كلية العلوم البنات

قسم علوم الحاسبات

دمج الصفات البايومترية المتعددة للانسان بأستخدام تقنيات التعلم العميق

رسالة مقدمة الى

مجلس كلية العلوم للبنات-جامعة بابل

وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

حوراء عبدالكريم حسين

بأشراف

أ.د حوراء حسن عباس

١٤٤٥هـ ٢٠٢٣م