

A COMPONENT BASED FACE RECOGNITION TECHNIQUE USING ONE SHOT LEARNING FOR FORENSIC APPLICATION

BY

YISHAK AMAN JIMJIMO



OFFICE OF GRADUATE STUDIES

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

ADAMA, ETHIOPIA

JANUARY 2021

A Component Based Face Recognition Technique using One Shot Learning for Forensic Application

By: Yishak Aman Jimjimo

Advisor: Dr. Mesfin Abebe



A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfilment for the Degree of Masters of Science in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University (ASTU)

Adama, Ethiopia

January 2021

Declaration

I hereby declare that this MSc thesis is my original work and has not been presented as a partial degree requirement for a degree in any other university, and that all sources of materials used for the thesis have been fully acknowledged.

Name: Yishak Aman

Signature: _____

This thesis has been submitted for examination with my approval as a thesis advisor.

Name: Mesfin Abebe (PHD)

Signature: _____

Date of Submission: _____

Acknowledgement

First and above all, I would like to thank to the almighty of God and St. Marry whose eternal and undying love kept me and for infinite mercy throughout my life and during this thesis work.

Next, My grateful thanks to my respected advisor Dr. Mesifn Abebe for his kindly approach, inevitable support, understanding and critical concerns, inevitable comment started from the proposal up to accomplish of this thesis work.

Next, I would like to express my gratitude to Anteneh Tilaye who assisted me to have broad knowledge in my research area and I would like to thank Yewediyanesh Tesfaye, Leoul Mesfin and Turegn Fikre for all your support.

Finally, I must express my very profound gratitude to my mother Asres Alemayehu, my father Aman Jimjimo, my sisters Hanan and Fana, and especially for my brother Eliyas being there for me in every aspect of my life, and for the support and continuous encouragement during my study and through the process of writing this thesis.

Table of Contents

Acknowledgement	d
List of Figures	v
List of Tables	vii
List of Abbreviations and Acronyms	viii
Abstract	x
Chapter One	1
1 Introduction	1
1.1 Background of the Study.....	1
1.2 Motivation of the Study.....	3
1.3 Statement of the Problem	4
1.4 Research question.....	5
1.5 Research Objective.....	5
1.5.1 General Objective	5
1.5.2 Specific Objectives	5
1.6 Scope and Limitation of the Study.....	5
1.7 Significance of the Study	6
1.8 Organization of the Thesis	6
Chapter Two.....	7
2 Literature Review and Related Work	7
2.1 Biometrics Techniques	7
2.2 Face Detection Methods.....	9
2.2.1 Feature-Based Face Detection Approach.....	10
2.2.2 Model-Based Face Detection Approaches.....	11
2.3 Face Recognition Approaches.....	11
2.3.1 Holistic Based Face Recognition Approaches	14
2.3.2 Feature Based Face Recognition Approach.....	15
2.3.3 Hybrid Based Face Recognition Approach.....	15
2.4 Facial Landmark based Face Recognition	16
2.5 Deep Learning based Face Recognition.....	17
2.6 Challenges in Face Recognition.....	18
2.7 Related Works	20

2.8	Summary of Related Work.....	22
Chapter Three.....		24
3	Research Methodology	24
3.1	Literature Review	25
3.2	Research Design.....	25
3.2.1	Problem Identification	25
3.2.2	Defining Objectives of the Study.....	26
3.2.3	Designing and Developing Prototype	26
3.3	Data Collection.....	26
3.3.1	GTAV Face Database	26
3.3.2	Self-Prepared Database	27
3.4	Multi-task Cascaded Convolutional Neural Networks (MTCNN)	28
3.5	Feature Extraction for Face Recognition Selection.....	29
3.6	MobileNet Convolutional Neural Network.....	30
3.7	One-Shot Learning	31
3.8	Development Tools and Packages.....	32
3.8.1	OpenCV-Python.....	32
3.8.2	TensorFlow	33
3.8.3	Keras	33
3.8.4	Numpy.....	33
3.9	Evaluation Metrics	34
Chapter Four		35
4	Proposed Component Based Face Recognition Technique	35
4.1	Overview of the Proposed Technique	35
4.2	Pre-Processing of Image for Face Detection.....	37
4.2.1	Collecting Face Image (Get face image datasets).....	37
4.3	Face and Facial Landmark Detection.....	38
4.3.1	Face Alignment.....	40
4.4	Feature Extraction (Feature Representation).....	40
4.4.1	Feature Extraction (Feature Representation) using MobileNetv2	40
4.5	One-Shot Model	41
4.6	Evaluation.....	42

4.7	Database	43
Chapter Five.....		44
5	Implementation of the Proposed Face Recognition Technique	44
5.1	Overview	44
5.2	Experiment and Development Setup.....	44
5.2.1	Working Environment	44
5.3	Dataset Preparation and Description.....	45
5.4	Face detection and Facial Component Extraction Implementation	50
5.4.1	Face Detection	51
5.4.2	Facial Component Extraction	51
5.5	Face Recognition Implementation	52
5.5.1	Facial Feature Extraction Implementation.....	52
5.5.2	One-Shot Implementation.....	53
5.6	Evaluation Implementation	55
Chapter Six.....		56
6	Evaluation, Results and Discussion of the Study	56
6.1	Overview	56
6.2	Face Detection.....	56
6.3	Face Recognition.....	58
6.3.1	Feature Extractor.....	58
6.3.2	One-shot Model Result	60
6.4	Output of the Study	69
Chapter Seven		70
7	Conclusion and Future Work.....	70
7.1	Conclusion.....	70
7.2	Future Work	71
References.....		72
Appendixes		77
Appendix I. Confusion Matrix for Feature Extractor and One-shot Model.....		77
Appendix II. Confusion Matrix for One-shot Model for Self-prepared Dataset.....		79
Appendix III. Evaluation Confusion Matrix Implementation.....		80
Appendix IV. Data Loader and Generator Implementation.....		81

Appendix V. Training Accuracy and Loss Plot Implementation 86

List of Figures

Figure 2. 1 Different Approaches to Face Detection [21]	10
Figure 2. 2 A hierarchy of face recognition approaches [16] [22].....	14
Figure 2. 3 Face Image with Pre-Specified Facial Landmarks Grouped in Region Subsets [27]	17
Figure 2. 4 Pose Variation	19
Figure 2.5 Illumination Variation [29].....	19
Figure 2. 6 Partial Face Image Variation [29].	19
Figure 2. 7 Facial Expression Variation [29].....	20
Figure 3. 1 Research Methodology	24
Figure 3. 2 Example of a Person ID on the Database (Different Pose Views)	27
Figure 3. 3 Example of a Person ID on the Database (Different Illumination)	27
Figure 3. 4 Example of a Person ID on the Database (Different Facial Expression)	27
Figure 3. 5 Example of Persons on the Self-Prepared Database.....	28
Figure 3. 6 The Architectures of P-Net, R-Net, and O-Net [39]	29
Figure 3. 7 Benchmark Analysis of Deep Neural Network Architectures[40]	29
Figure 3. 8 Architecture of MobileNet [34].....	31
Figure 3. 9 Siamese Network Architecture	31
Figure 3. 10 Deep Learning Framework Power Score 2018 [44].....	33
Figure 4. 1 General Face Recognition Process Flow [47]	36
Figure 4. 2 Proposed Approach for Face Recognition.....	36
Figure 4. 3 Pre-processing steps for Face Recognition.....	37
Figure 4. 4 Example of Face and Landmark Detection	39
Figure 4. 5 Feature Extractor using Mobile Net	41
Figure 4. 6 The Proposed One-shot Model Architecture	42
Figure 5. 1 Sample Detected Faces on GTAV database for Augmentation	45
Figure 5. 2 Sample Detected Faces on Self-prepared database for Image Augmentation.....	46
Figure 5. 3 Code Sample for Horizontal Flip Augmentation.....	46
Figure 5. 4 Sample Images of GTAV database after Horizontal Flip Augmentation.....	46
Figure 5. 5 Sample Images of Self-prepared database after Horizontal Flip Augmentation.....	47
Figure 5. 6 Code Sample for Brightness Augmentation	47
Figure 5. 7 Sample Images of GTAV database after Random Brightness Augmentation	48

Figure 5. 8 Sample Images of Self-prepared database after Random Brightness Augmentation.	48
Figure 5. 9 Code Snippet for Gaussian Blur Augmentation.	49
Figure 5. 10 Sample Images after Gaussian Blur Augmentation on GTAV database.....	49
Figure 5. 11 Sample Images after Gaussian Blur Augmentation on Self-prepared database	50
Figure 5. 12 Sample Code for Initializing the Face Detection Module and Face Detection	50
Figure 5. 13 Sample Face and Facial Feature Detection Output	50
Figure 5. 14 Code Snippet for Detected Face Extraction	51
Figure 5. 15 Code Snippet for Facial Component Extraction.....	51
Figure 5. 16 Code Snippet for Loading MobileNetV2	52
Figure 5. 17 Code Snippet for Set Trainable Parameters.....	52
Figure 5. 18 Code Snippet for Facial Component Feature Extractor	53
Figure 5. 19 Code Snippet for Model Construction Feature Extraction	53
Figure 5. 20 Code Snippet for Siamese Network Input Layers Initialization.....	53
Figure 5. 21 Code Snippet for One Shot Model Construction.....	54
Figure 5. 22 The Implementation of One Shot Architecture using Siamese Network	54
Figure 5. 23 Code Snippet for Euclidean Distance Implementation as Loss Function	55
Figure 5. 24 Code Snippet for Matrix Creation	55
Figure 5. 25 Code Snippet for Recall, Precision, Accuracy and F1 Score	55
Figure 6. 1 Feature Extractor Accuracy and Loss during Training	59
Figure 6. 2 Confusion Matrix of the Feature Extractor on Test Set	60
Figure 6. 3 One-shot Training Accuracy and Loss on Training and Testing Set	61
Figure 6. 4 Confusion Matrix for One-shot Model in GTAV database.....	61
Figure 6. 5 Confusion Matrix for One-shot Model in Self-prepared database	62
Figure 6. 6 One-shot Training Accuracy and Loss of OS - AE.....	64
Figure 6. 7 One-shot Training Accuracy and Loss of OS + AE	64
Figure 6. 8 One-shot Training Accuracy and Loss of OS – Comp.....	65

List of Tables

Table 2. 1 : Related Works on Face Detection and Recognition	23
Table 5. 1 Hardware and Software Specification of the Machine	44
Table 6. 1 Sample Face Detection and Facial Components Images in GTAV database	57
Table 6. 2 Sample Face Detection and Facial Components Images in Self-prepared database ...	58
Table 6. 3 One-shot models used for Experiment.....	63
Table 6. 4 Comparison of Different One-shot Models	65
Table 6. 5 Related Work comparison with Proposed Model	66
Table 6. 6 Sample Images of Face Recognition in GTAV database	67
Table 6. 7 Sample Images of Face Recognition in Self-prepared database.....	68

List of Abbreviations and Acronyms

AAM	Active Appearance Model
AE	Auto Encoder
ANN	Artificial Neural Network
CBFR	Component- based Face Recognition
CNN	Convolutional Neural Network
Comp	Component
Colab	Collaborator
DCT	Discrete Cosine Transform
DNN	Deep Neural Network
ELM	Extreme Learning Machine
FR	Face Recognition
GPU	Graphics Processing Unit
HSV	Hue Saturation Value
ICA	Independent Component Analysis
LDA	Linear Discriminant Analysis
MTCNN	Multi- Task cascaded convolutional Neural Network
ORL	Olivetti Research Laboratory
OS	One Shot
O-Net	Output Network
PCA	Principal Component Analysis
PIE	Pose Illumination and Expression
PICS	Psychological Image Collection at Stirling
PIFR	Pose Invariant Face Recognition

P-Net	Proposal Network
RBF	Radial Basis Functions
Res Net	Residual Convolution Neural Network
R-Net	Refine Network
TPU	Tensor Processing Unit

Abstract

Face recognition is a technique of verifying or identifying the identity of a person by their face and it is one of the significant issues in object recognition and computer vision. It is an important and regular forensic tool used by criminal investigators. An essential part of human beings is the face and requires detection and recognition for various applications such as security and forensic investigation. However, due to the difficulties of different facial expressions, position variations, occlusion, aging, and resolution in image or video sequencing images, it requires proper techniques for face detection and recognition. Hence, the systems working on components of the face or facial landmarks have gained great importance. Due to the variety of criminal activity, accurate and effective identification has become a basic requirement for forensic application. Forensic examiners differentiate different facial areas of face images acquired from uncontrolled and controlled conditions captured from the suspect. This research focuses on Facial-landmark based Forensic Face Recognition. This study proposes a method by using Component-based Face Recognition with the information acquired from component of faces, by combining MobileNet-v2 with a Siamese Neural Network to increase the accuracy of face recognition rate with faces of different pose variation, expressions, and illuminations. The experiment results show that our proposed framework of face recognition by different facial expressions, poses variations, and illumination image data on the GTAV database of over 44 persons, this study have achieved a recognition accuracy rate of 95.4%. The Self-prepared face database obtained a recognition accuracy rate of 93%. This Paper gets better performance compared to face recognition without components, One-shot with Autoencoder, and One-shot without Autoencoder.

Keywords- Forensics, MobileNet-v2, Siamese Neural Network, Component-Based, Hybrid Face Recognition, Criminal Investigators

Chapter One

1 Introduction

1.1 Background of the Study

The human face is the first view of attention in society, playing a significant role in conveying identity and emotion. Face Recognition is the process of verifying or identifying the identity of a person by using their face [1], it is to look at the major features of the face and compare these to an equivalent feature on other faces. Face recognition (FR) technology identified as an active area of research in recent years due to the increase in security demands and therefore the potential of the technology in law enforcement and commercial use. Face recognition systems can be used to identify people in images, videos, or in real-time moreover, it is used for strengthening the legal system for example by data protection, security law-enforcement to increase activity to combat crime and terrorism. It can be applied to diagnose diseases that cause noticeable changes in appearance and Facial recognition can aid forensic investigations by automatically recognizing individuals in security footage or other videos.

The problem of face recognition is complex and highly challenging because it's affected by a spread of parameters, including illumination, pose orientation, expression, head size, image obscuring, and image background [1][2]. In our entire life, we can remember hundreds or even millions of faces and recognize a face in various variations of perspective, illuminations, ages, etc. Under very poor illumination conditions, a face can still be recognized, because the position of the various facial features are usually sufficient for recognition.

Most of the proposed methods for face recognition deal with 2D appearances, such as Eigenfaces [3] and, Fisherfaces [4]. However, such methods are the most sensitive to varying illumination and pose. Face recognition can be done by either verification or identification. Verification involves a comparison of the features related to the given face with the corresponding feature template stored in the database against a claimed identity. It involves only one single comparison. Identification, however, compares the given face with all the feature templates stored in the database to find the identity of a face among several possibilities that match the most. This is why identification takes time proportional to the size of the database [5]. Identification involves extracting information like landmarks, expression, pose, age, gender, identity, etc. It has applications that include law enforcement, active user authentication, facial payment biometrics, self-driving cars, etc. Face recognition is the ability to determine a subject's identity based on his facial characteristics.

In forensic examiners, the face is the focus of attention, playing an important role in conveying identity, and it has become a useful and routine forensic tool used by criminal investigators. According to recent studies, Automatic face recognition methods aim to identify a particular person using full-face images. Forensic face recognition is more complex than automated face recognition because facial images collected in non-ideal situations must be able to be controlled. The performance of these face recognition methods affected by various factors such as a change in pose, expressions, and illuminations. An individual classifier based on the different appearance of face image has a different sensitivity to these variations, which encourages going towards a hybrid approach i.e. combination of various face recognition methods, to enhance accuracy [1] [2].

Methods of Component-based Face Recognition (CBFR) can be of significant help in resolving changes in position, expressions, and illuminations. CBFR is recognition focused on the forehead, lips, chin, eyes, nose, mouth, ears, eyebrows, or face of specific sub-regions. The basic idea of the CBFR technique is to compensate for changes in the pose by maintaining the size and shape of the individual facial components. Illumination changes only affect a certain part of the face, so the CBFR helps to increase accuracy [1].

Today, the face recognition system relatively established in well-controlled environments. As such, machines are designed with unavoidable user cooperation protocols to capture the maximum information of the human face for the preeminent recognition performance. However, these protocols have accentuated their deficiency under certain constraints because some individuals are reluctant to provide their full face for recognition. For example, individuals who wear face masks, sunglasses, and veils are restricted to provide their full face due to occupation, contagious disease, privacy, and religious practices [6]. Therefore, a component-based face recognition system based on users' preferences could be one of the potential solutions to solve these conflicts.

At present, the face recognition technology has been quite mature under the state of controllable lighting and little intraclass change. However, the performance of face recognition in the non-ideal condition still needed to be improved. PIE problem [7] is the non-ideal condition that face recognition should solve especially the problem of variable illumination, pose variation, and expression. This study proposed a way that supported invariant features, which used the features of the face image that didn't vary with the change of lighting conditions to process, that is, to find the light insensitive features [8][9][10]. To achieve facial recognition, many types of features extracted from images can be used, including soft biometrics such as gender, facial landmarks, ethnicity [11].

According to recent research, soft biometric traits can be correctly combined with classical facial recognition methods to increase the accuracy of the verification or identification process, despite not being entirely discriminatory on their own. In particular, facial marks have become extremely useful: they can be used to narrow down the search of identity, while not identifying a person uniquely [12].

Face recognition is still a challenge for recognizing a face in motion images, twin's, pose variations, having different accessories like a beard, glasses, hair color, hair partition, make-up, different facial expressions under different illumination condition, noise, light intensity, occlusion, and thermal image for face matching and error generation. So this study proposed a new method by combining MobileNetv2 and Siamese Neural Network using the information obtained from Facial landmarks to increase accuracy and this method is capable of handling light variation, facial expression and pose variation. As a result, increasing the robustness of existing systems in the face of various challenges is the aim of this present research.

1.2 Motivation of the Study

Face recognition is a technique of identifying or verifying an individual's identity using their face or comparing a face against a database of faces from a digital image or a video frame. Face recognition is challenging, one of the major challenges, when the dataset is limited, is that an individual's face may look different if in various position, lightning, and expression.

The motivation behind this work is the place where I was grown, criminals who have committed crime tried to flee or escape to other places to hide when police looking them. In addition, they change their names or identities, and after a time they came back to their previous place to live as they did nothing, however, the police never identify them. Even if they did another crime, their previous crime case will not reopened.

Therefore, criminal is aware that there are no ways of making his or her identity known and these become the most necessary research areas because of their benefit in various applications such as in security systems, Identification systems, and crime investigation. The increasing need for security in public and private sectors conducts this work and in various scenarios of forensic applications. In this study, facial landmarks are helpful information and this information can be improving the overall face-recognition accuracy. Face recognition technology has enhanced over the years but still there are some gaps. Achieving a face recognition technology that can recognize humans will certainly be a lion's share against the emergence of crime in our country by providing a set of facial features that can be understood by non-experts like judges and to support the work of forensic examiners.

Generally, the gap or problem that discussed in the statement of the problem section initiated me to focus on the area of Face Recognition.

1.3 Statement of the Problem

Face recognition is the science deals with how the biological systems recognize different faces and how to be emulated by computer systems. It needs proper techniques for face recognition with problems of different pose variations, expressions, and illuminations [13]. In a surveillance system, the camera usually mounted to a location where the people cannot reach the camera. Mounting camera, a high location, the faces viewed by some angle degree. This is an easy case in city surveillance applications. The following and the most difficult case is that people naturally go through the camera view. They are doing not even check out the camera lens. Authorities cannot restrict people's behaviours publicly places [14].

The difference between two images of an equivalent subject under the different view angles is bigger than differences between images of two different subjects under an equivalent view. Recognition in such cases must be complete in an accurate way. Hence, Pose variation [6] [15] [16] is one of the major sources of performance degradation in face recognition it is essential for non-cooperative subjects and in an uncontrolled and less restricted environment. The problem of face recognition occurred when the system used a rotated testing image to confirm an individual identity. The frontal face has more information about an individual than the posed face. When a face captured with different poses, it is difficult to recognize with the training set of frontal faces in partial face recognition.

Illumination variation [15] [16] is another challenge that affects the performance of face recognition tasks. This variation caused by various lighting environments and mentioned to possess a larger appearance difference than the difference caused by different identities. One of the most difficult problems that a practical face recognition system needs to achieve is the impact of variation within the lighting conditions, which causes significant changes within the facial appearance. More specifically, due to the change in face identity, the difference between the images of an identical face is usually greater than image differences due to lighting and viewing direction. If the change appears on the image caused by illumination is larger than the difference between individuals, the face recognition system would not be able to recognize the input image.

Face expression [15] [16] is a less important issue compare with pose variation and lighting but it affects the face recognition results. Human shows their feelings with different facial expression such as happiness, sadness, anger, contempt, disgust, fear, joy, and surprise. The expression variation leads to not only the position change but also the facial feature shape change.

Although an in-depth eye or smiling face does affect the recognition rate. Many research papers specialize in small changes on the face surface. However, huge changes in expression are still an unsolved problem. Therefore, the proposed methods can be solving help in overcoming these challenges and increase the performance.

1.4 Research question

In this study, the researcher attempt to answer the following research question.

- RQ1: What are the most effective methods for forensic face recognition system?
- RQ2: Is it possible to boost face recognition accuracy by using Facial Landmarks?
- RQ3: Can Facial Landmark-based face recognition solve pose variation, expression, and illumination problems?

1.5 Research Objective

1.5.1 General Objective

The main objective of this study is to develop a Component-based Face Recognition technique using MobileNet-v2 and Siamese Neural Network that handle illumination, pose variation, and facial expression and to increase the accuracy of forensic face recognition.

1.5.2 Specific Objectives

- Determine the major vulnerable conditions that affect the face recognition process
- Identify the major problems of image that influence the process of Face Recognition.
- Select an appropriate algorithm and collecting dataset for training and testing.
- Developing the Component based Face Recognition model
- Testing the model with a test set to find the best parameters.
- Evaluate the performance of developed model.

1.6 Scope and Limitation of the Study

This research includes the design and development of face recognition on forensic application. In this study pose variation, illumination, and facial expression were all taken into consideration. The database used excludes partial facial images and mixed variations.

Due to its performance, this study analyze component-based facial recognition during enhance the accuracy of the face recognition process.

The limitations of this study is related with data-set properties, it contain 2D image-based representation. Despite the high success rate achieved in 2D face recognition but, 2D images have

several limitations, including image magnification and distortion, superimposition of anatomical structures, during the limited number of 3D image data sets , 3D images were not included in this analysis.

1.7 Significance of the Study

The result of the research is primarily beneficial for Forensic examiners by avoiding manual examination of facial images for a match with huge database of mugshots. In general, the outcome of this research has introduces robust technology to our country criminal investigation Process.

Some of the applications where this could be applied in Law Enforcement Agencies in Crime stopping and suspect alert with ability to search and identify suspects quickly, in Police Clearance Certification, in Airport Security Systems and in Face Identification.

1.8 Organization of the Thesis

The thesis is organized into Seven Chapters:

Introduction (Chapter 1) describes the background of the study and the motivation, the statement of the problem and the research question to be answered, the methodology to achieve the objective of the study, the scope and limitations, and the application of the result.

Literature Review and Related Work (Chapter 2) gives an overview of computer vision, face detection and recognition, the challenge in face recognition, different techniques of face recognition, previous works on face detection and recognition for forensic application, and comparison of different approaches taken by related works.

The methodology of the study (Chapter 3) describes methodology in the study in detail. Which includes literature review, , data collection method, proposed system design method, proposed model implementation method, proposed model evaluation, development tools, and packages.

Proposed Component Based Face Recognition Technique (Chapter 4) presents the face detection and recognition algorithm and deals with the proposed model experimental setups, procedures, and evaluation metrics.

Implementation of the Proposed Face Recognition Technique (chapter 5) discusses the implementation step of the proposed solution, which contains training Steps, testing steps with implementation, sample source code of the proposed solution.

Result, Evaluation, and Discussion (chapter 6) present the result and discussion of the face recognition model and the result discussed by comparing it with other related works.

Conclusion and Future Work (Chapter 7) describes the conclusion and feature work of the study.

Chapter Two

2 Literature Review and Related Work

This chapter reviews some common known biometric systems formed on the orientation, shape, and size of facial traits, which can be regarded as a soft biometric approach. Special attention was given to existing face recognition methods and tools that can be used to develop an accurate and efficient system.

2.1 Biometrics Techniques

The word biometric comes from two Greek words, “bio” and “metrics” where bio means that “life” and metrics means that “biological measurement”. The combination of these two words gives meaning “life measurement” [12][17]. The biometric system is an identification mechanism that allows people to be automatically recognized based on their specific physiological or behavioral features.

Biometrics is a mechanized technique to perceive an individual, identifying a person digitally to grant access to systems, devices or data, to securing different environments and in other related security works.

The earlier of biometrics incorporates the specific evidence of peoples by special body elements, some unique features that differentiate them from other individuals like skin color or texture, retina, voice, height, etc. Examples of these biometric identifiers are fingerprint, facial patterns, hand-geometry, retina, palm print, voice, gait, signature, and keystroke dynamics, which are widely used for the task of person recognition [12][16].

Behavioral characteristics are related to the patterns of behavioral, natural aspects of the human being such as signature, voice, keystroke, ear, giant, etc. [16]. Whereas physiological characteristics deal with the physical appearance of the human body such as the face, finger, iris, etc. Biometric techniques of human identification have earned great attention recently, mostly because they simply deal with most troubles of traditional identification.

In biometric human identification systems, users are specified not by something they need to have remembered or carry with them like a password, they specify by who they are. Biometric Methods offers crime finding by associating the traces to the persons saved in the database, evaluating the identity of persons, and choosing a subdivision of persons from which the trace could originate [15]. Biometric technology is a system for pattern recognition that receives physical or behavioral information from a person, extracts a prominent feature set from the data, differentiates this feature set against the features set saved in the database and gives the outcome of the contrasting.

This section discussed some commonly used biometric method for choosing the best method among them. It is possible to operate all biometric systems in either verification or identification mode Face verification requires a direct match that compares an image of the query face with an image of the registration face whose identity claimed. Identification of faces requires several matches that differentiate a query face from multiple faces in the enrollment database in order to identify the identity of the query face with one of those in the database.

Some of biometric systems are face recognition, hand geometry, voice, and fingerprints. These technologies are more reliable compared with other methods such as password and RFID because it eliminates manual input or identity card exchange. Every biometric has a strength and a weakness of its own. Therefore, choosing a typical biometric depends on different parameters such as performance, uniqueness, acceptability, and collectability. In forensics, biometric recognition can be used in two separate ways as a method to aid in the investigation and support the information given in a court of law [18]. It should remembered that these two use-cases have very distinct specifications.

In the first case, the accuracy and speed of the biometric system under demanding data conditions are the main requirements. However, low levels of errors created by the system are tolerable during this situation the investigating officers can make use of other contextual information (e.g. race of the suspect age, and gender) to eliminate some of the errors.

The primary requirement in the second case is the scientific presentation to a court of law of biometric proof with a clear statistical basis.

As the biometric process is usually a method of pattern-recognition, the need for data acquisition, pre-processing, data representation, and decision-making is needed. This shows the extraction of physiological or behavioral characteristics of people. Earlier biometric using it as a template database to be used after for recognition. Biometrics are considered unique to the person, and they may be used in variety to ensure more excellent accuracy of identification. Soft biometric traits to enrich the identity information offered by the primary biometric identifiers and also the description of soft biometric traits were presented as characteristics that provide some details or characteristics about the person, but lack the differentiation and permanence to sufficiently compare any two individuals. [5][17].

Some of the common characteristics of humans that can be regarded as soft biometrics are Gender, height, weight, age, and ethnicity [16]. According to the concept of soft biometric marks, some information about the person may also be provided by characteristics, tattoos, and scars and the information may not be sufficient to identify human beings.

2.2 Face Detection Methods

Face detection is a computer vision technology that defines the sizes and location of human individual faces in images or videos regardless of the identity of that face [19]. In the history of face detection and recognition, different approach and methods are used to increase the accuracy, quality, and, efficiency of an algorithm such as PCA, LDA, and ICA are Holistic methods widely used for automatic feature extraction and recognition. Gabor is the feature-based approach and AAM is the hybrid approach has solved some issues of face recognition. However, the processing speed is still challenging when applying for real-time applications. Various methods to Face Recognition (FR) have been proposed to resolve the variations that occur with the FR process, Machine Learning [2], Template Matching [5], Information Theory [7], Geometrical Modelling [16], Hough Transform [15], Motion Extraction [20], and Color Analysis are examples of these methods.

In the related literature, several CBFR techniques use a type of components and approaches for face recognition and feature extraction [1] confirmed that the human visual system is naturally able to recognize faces from isolated features based on the information. Verified that face processing is the result of combining individual face component-based processing by comparing the optimal Bayesian integrator with the human part integrator, indeed, any human being identifies a face by combining individual facial features [5].

The authors stated in [14] that early face detection activity moved as quickly as at the beginning of the 1970s, where the basic heuristic and anthropometric systems were used. Due to various assumptions, for instance, plain foundation, frontal face, a typical visa image situation, these structures are usually rigid. Human face recognition is not only about face recognition at all, it is also more about face finding or face detection. The primary step in automated facial recognition was shown to be the accurate detection of human faces in optional scenes, is that the most significant process included. When faces can be placed correctly in any scene, the recognition process then is not so complicated anymore.

Face detection is the initial level of face recognition as it automatically detects a face from a complicated background to which the face recognition algorithm can use, but detection itself involves many complications like poses, background, illumination, etc. [11] [21]. Given an arbitrary image, as presented in [22], the objective of face detection is to decide whether human faces appear in a given image or not and, if present, return the image size and location of every face. Practically, face detection algorithms will aim at generating bounding boxes (often rectangular or elliptical) around all the faces in the image and only around faces. Face detection is an initial step in any face recognition system that able to detect facial features and ignores other image components, such as cars, buildings, trees, and

bodies [21]. The human face can be detect from on-line surveillance or off-line image. In face detection, facial features such as eye, nose, mouth, cheek, etc. are the main component to be detected. Face detection techniques require prior face information, so they can be organized efficiently into two large categories that can be used to face information from one person to another. The first approach is feature-based approach, while the second approach is Model-based approach. The Figure 2.1 below shows face detection approaches.

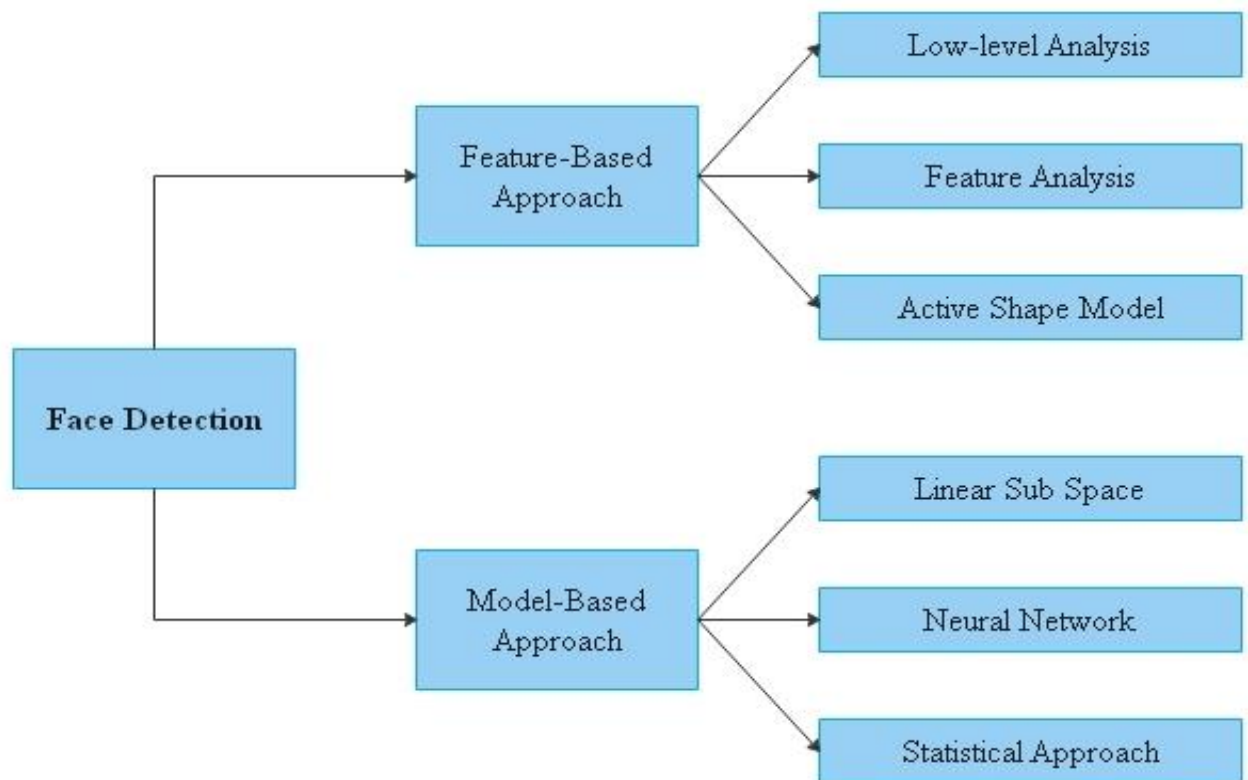


Figure 2. 1 Different Approaches to Face Detection [21]

2.2.1 Feature-Based Face Detection Approach

The development of a feature-based approach can be further divided into three subcategories as shown in figure 2.1. This approach applies several techniques in face detection that make explicit use of face information and follow the classical detection methodology in which low-level features are obtained prior to knowledge-based analysis. Generally, in these techniques Face Detection tasks are accomplished by manipulating angles, distance, and area measurements of the visual characteristics obtained from the image. The feature-based approach focuses on the segmentation of facial features using pixel properties such as edge, Grayscale, motion, and color information, and Feature Analysis [21]. One issue with this approach is that it is tough to translate our knowledge about-faces into

regulations effectively. If the regulations are strict it may fail to detect faces that don't pass all the regulations. If the regulations are too general, it may give many false positives [11].

In any Face Detection problem use a feature-based approach for locating the face firstly, it will start with low-level feature analysis in which segmentation of visual features applying pixel properties such as gray-scale and color will be done. After that, the characteristics generated from this are likely to be ambiguous, which can be solved by higher-level analysis of features such as the Template Matching approach that describes and uses face geometry information to characterize and subsequently verify different features from their ambiguous state [21][23].

2.2.2 Model-Based Face Detection Approaches

The model-based face recognition approach is aimed at constructing a model of the human face, which is able to capture the facial variation. The prior knowledge of the human face is highly utilized to design the model and further, it can be divided into three subcategories as shown in figure 2.1. Feature-based matching, for example, derives distance and relative position characteristics from the positioning of internal facial parts.

2.3 Face Recognition Approaches

Face recognition is an activity common to humans, that the person does not even recognize the extensive variety of times it is performed every day. While research has been conducted since the 1960s on automated face recognition, it has only recently caught the attention of the scientific community. Many face analysis and face modeling techniques have progressed significantly in the last decade [19][24]. However, the trustworthiness of face recognition schemes still poses an outstanding challenge to the scientific community. FR contains two operational modes. First, verification mode is known as one-to-one matching in biometrics where the people claim an identity and given input image is matched up against that of the claimed identity by their several feature sets to validate the claim. The verification operational mode is used to select a face out of many faces in a face database to search out whether the face details belong to a selected person. Second, the identification mode is known as one-to-many matching where the people do not claim an identity; to find the best match and, thus, its identity, the input user image is compared with a group of labeled user images inside a database. Identification involves taking the individual and matching their characteristics to a database of available identities.

One of the few biometric techniques with the advantages of low intrusiveness and high accuracy is Face Recognition. In the absence of being intrusive, it has the accuracy of a physiological approach. A complete face recognition system includes face detection, face pre-processing, and face recognition processes [1] [2] [5] [19]. The face recognition system begins with detecting the presence of a face in

an image. Generally, a face detection system can decide if an image includes a face or not. If it does, the system's task is to locate the position of one or more faces within the image. After all, this process becomes hard if variations in light, facial smiling, sadness, surprise, etc.), orientation position and morphological criteria (glasses, mustaches, etc.) occur. All of those barriers will stop the correct face detection and consequently decrease the rate of face detection. In a regular face recognition scenario, face images from a variety of subjects are enrolled into the system as gallery data, and therefore face image of a test subject is paired to gallery data using one-to-one or one-to-many scheme. There are three different modalities that are used in face recognition applications: 2D, 3D, and video.

In 2D image-based face recognition systems, an image of a user obtained and related to a gallery of snapshots to determine a human identity.

In this procedure, the user is looking to be cooperative, and supply a frontal face image under uniform illuminating conditions with a simple background to enable the capture and segmentation of a high-quality face image. However, it is currently well known that small variations in lighting and position can drastically degrade the performance of the single-shot 2D image-based face recognition systems. Some of the well-known algorithms for 2D face recognition are based on Principle Component Analysis (PCA) [6][24], and Linear Discriminant Analysis (LDA) [6].

3D face recognition techniques use the surface geometry of the face. In contrast to 2D face recognition, 3D face recognition is powerful against illumination and position variations due to the invariance of the 3D shape against these variations. Many approaches have been developed based on reconstructed 3D models from a collection of 2D images [24]. The reconstructed 3D model used for several 2D projection images that are identical to probe images. On the other hand, to create a frontal view of the probe image with lighting conditions and arbitrary pose, the reconstructed 3D model can be applied; the recognition is achieved by comparing the synthesized probe in a frontal position.

Facial recognition systems typically contain four steps, face detection, face preprocessing (light correction, face alignment/normalization, and etc.), feature extraction, and feature matching [19] [14]. Within the given images, Face Detection is used to identify faces, and it is the first and essential step in face recognition. In addition, face recognition refers to the psychological mechanism by which human discover and view faces in a visual scene. Pre-processing is a general term for image processes, all of which are input and images of output intensity, at the lowest abstraction level. The aim of pre-processing is to enhance image data that removes undesirable distortions or enhances certain image characteristics required for further processing. Feature Extraction is a special type of reduction of dimensionality by which the first raw data set reduced to more controllable processing classes.

A component is a big number of variables that need a lot of computational resources to deal with are a part of such big data sets. It starts from the first set of measured data and produces derived values or features intended to be non-redundant and insightful, helping to learn afterward and generalization steps, and in some situations, leading to best human interpretations and it is relevant to dimensionality reduction.

Feature Matching is an effective technique in a cluttered scene to detect a particular target [14]. Instead of different objects, this technique detects single objects. The algorithm is set up to evaluate and compare the correspondence of points between the source image and the target image.

Face Recognition is used for different purpose and can summarize in to three primary face tasks [16] [19].

The first one is Verification (authentication) the task of comparing a candidate's face to another and verifying whether it is a match. It is a one-to-one mapping: you have to check if this person is the correct one; the second is Identification (recognition) the task is mostly aimed at applications not requiring user interaction. The identification test works from the assumption that all faces in the test are of known persons, the last is watch list the task is a generalization of the identification task which includes unknown people.

The basic question relevant to face classification is that what type of structural code should be considered to achieve face recognition. To address these recognition challenges for machine identification of human faces a number of algorithms have been proposed. These algorithms can be categorized into three main categories: The holistic approach, the Feature-based approach, and the Hybrid approach.

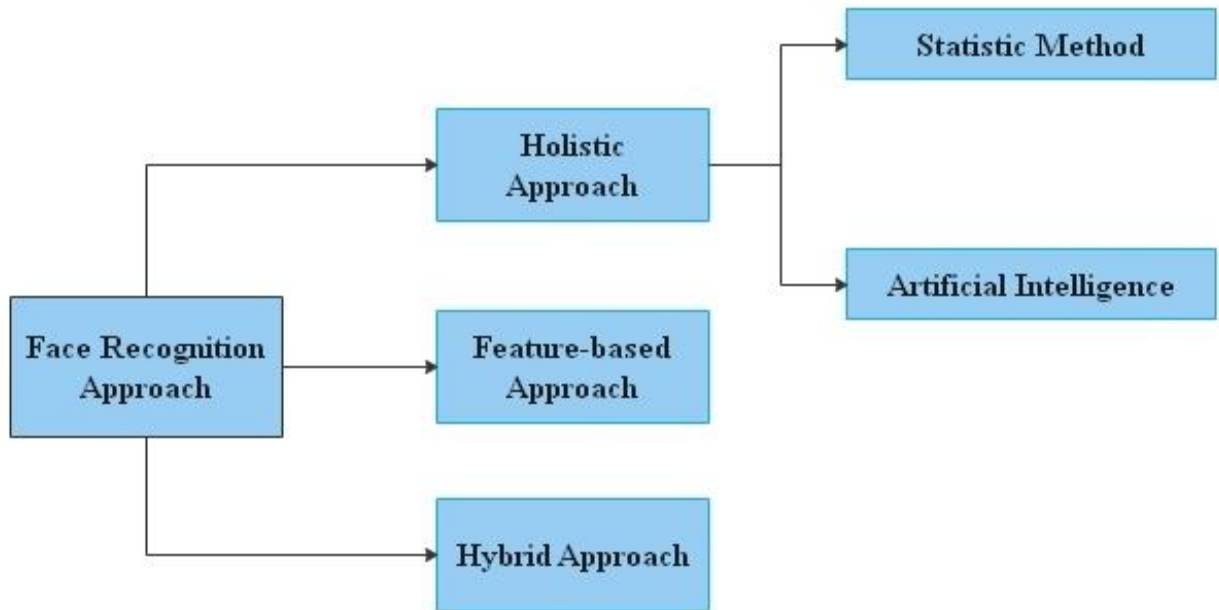


Figure 2. 2 A hierarchy of face recognition approaches [16] [22]

2.3.1 Holistic Based Face Recognition Approaches

The holistic approach utilizes global information from faces to perform face recognition. The global information from faces is basically detailed by a small number of features, which are directly derived face images from the pixel information. These small numbers of features clearly capture the distinction between different faces and are thus used to classify individuals uniquely [25]. While holistic approaches are based on global features, feature-based ones are built upon local facial features extracted from local components such as eyes, nose, mouth, etc., and local segmented regions. Figure 2.2 shows a different approach of face recognition using a holistic approach. One of the best instances of holistic techniques is Eigenfaces' most widely used technique for face recognition, Linear Discriminant Analysis, Principal Component Analysis, and independent component analysis, etc.

Statistical Based Face Recognition Approach: The face image density is calculated and the density set values are compared with the density values of database images [22]. Purely statistical approach depending initially on local histogram analysis and absolute gray-scale values. This calculation is very valuable and directly suffering below the usual gaps pathways, like face orientation scaling and light variations.

Artificial Intelligence Approach: Artificial Intelligence approach with tools such as neural networks and automatically recognizes the faces of learning techniques [22].

2.3.2 Feature Based Face Recognition Approach

Opposite to holistic, feature-based face recognition uses a priori information or local features of faces to select a number of features to exclusively identify individuals. Local features have included the layout of the mouth, nose, eyes, chin, and head, selected from face images. The feature-based position considers that significant features such as noses, eyes, and mouth play more roles in describing and remembering a person [14].

Typically, in these techniques, local features such as the mouth, nose, and eyes are first extracted and their positions and local statistics (appearance or geometric) are fed into a structural classifier. Objects are generally identified by their unique characteristics. There are several features within the human face, which can be recognized within the face and many other objects. By extracting structural characteristics such as mouth, eyes, nose, etc., it identifies faces, and then uses them to find a face. Commonly, some sort of statistical classifier qualified then supportive to divide between facial and non-facial regions.

This technique primarily based on the nose portion of the images acquired due to grayscale conversion and intensity transformation. Another study in which the face recognition is done with the guidance of edge information refined by the support of decreased sample size. Low dimension space for face images is done by Discrete Cosine Transform (DCT) [25].

In 2018, Rupali Sandip Kute [1] presented an approach to component-based face recognition and transfer learning association and shows that data obtained from full-face images is transferred to identify facial components. Although these face parts and also the face itself are from different domains, they share common information that is used to convey the knowledge earned from one domain to another.

In 2013, Kathryn Bonnen [5] proposed component-based representation offers higher recognition accuracies over holistic based representations. Furthermore, they show that the recommended component-based representations are improved recognition accuracy on the occluded face images and more powerful to changes in facial position, in forensic scenarios.

2.3.3 Hybrid Based Face Recognition Approach

The hybrid approach has a particular standing between face recognition systems as combined different recognition approaches in either serial or parallel to defeat the weakness of individual methods [26]. Hybrid face recognition systems are just as the human understanding system works both the full face region and local features to remember or to recognize a face, a machine recognition system should use both. One can argue that these methods could probably provide the best of the two types of methods.

In hybrid based face recognition, all individual face shares the same basic features arranged in some common pattern and each individual face can be differentiated by analyzing a face component and holistic information. Both the holistic and analytic face recognition approaches have their own strength and weakness, therefore by combining these approaches; a high-performance face recognition system can be achieved. Hybrid approach compared to holistic approaches and feature-based approaches have a significant advantage they can perform much better under various uncontrolled conditions [26][14].

Kanchikere [13] Proposed a face recognition system based on a hybrid optimized Kernel Extreme Learning Machine. The chief contribution of this work is that the initial input video is clipped into the frames, and then removes the noise from the image using MWF. Next, detect the pre-processed image using the Viola-Jones algorithm, and the features are extorted then the extorted features are given as the input to the Modified PCA for feature reduction. Finally, the lessened features are inputted to the classification using a hybrid optimized kernel ELM approach.

The authors in [25] used real-time face recognition by combining both holistic and analytic approaches. They used PCA to decrease the dimensionality of the data set and help to increase the accuracy of real-time face recognition. To increase the accuracy of the classifier, they execute a hybrid model that merges different classification techniques i.e. ANN and LDA. Both ANN and LDA are used for predicting data, but LDA only used with seen data. In addition, the LDA may not function with high dimensional data set, which means that the performance of the recognizer may be degraded.

Therefore, the PCA is used for feature reduction and select essential features, and then these features are used to train and test the LDA classifier. The usefulness of the hybrid model was evaluated by comparing the experimental result received from the data set using LDA and ANN. By combining the LDA and ANN, they reach a 91.2% accuracy rate.

2.4 Facial Landmark based Face Recognition

Facial landmark is a collection of key points on human face images, and they are used to localizing and represent salient regions of the face, like the nose, right eye, left eye, right mouth corner, and left mouth corner. Their real coordinates (x, y) on the image outline these points [27].

Face landmark prediction is when main known key points on a face from an image using face detection model with a noted collection of features or points that should appear on any face. Therefore, it moves those points around until they match the face image because the points are predefined, we should already know which of the points make up the mouse, the nose, the eyes, and so on.

The aim of using facial landmarks is to use shape prediction techniques to detect key facial structures on the face. There is a two-step process for Detecting facial landmarks:

- i. Positioning the face within the image.
- ii. Detect the facial landmark inside the face.

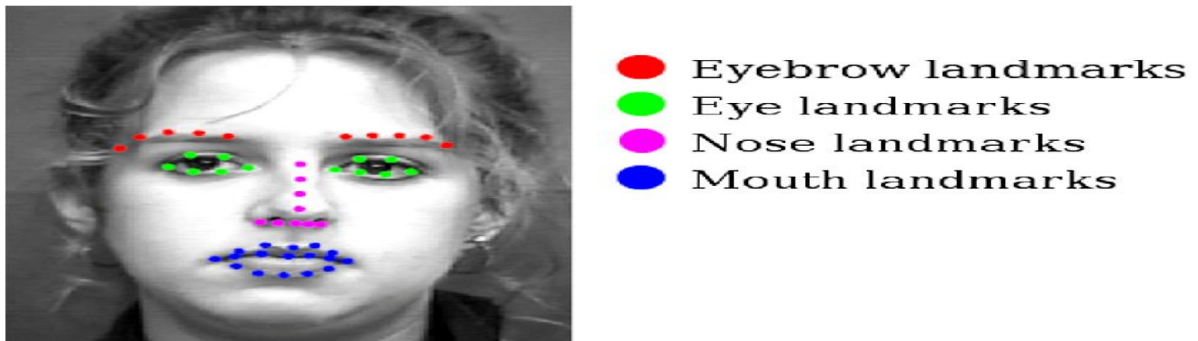


Figure 2. 3 Face Image with Pre-Specified Facial Landmarks Grouped in Region Subsets [27]

Facial landmarks are salient image features that express the color, shape, or texture of face images through simple data [27]. In facial landmark recognition, every landmark can be considered as a moving object within a local region of the input face image. In this scenario, every landmark is embedded into a non-overlapping background in a small fragment of the captured face image. Moreover, the input face images can be degraded by the sensor's noise, geometrical modifications, and non-uniform illumination.

Information regarding the location and motion of a collection of facial landmarks in a sequence of captured images is important for face analysis, as an example, in applications such as face recognition [28], facial expression recognition, and pose estimation [27].

2.5 Deep Learning based Face Recognition

Deep learning, also known as deep structured learning or hierarchical learning, belongs to machine learning methods, which are based on understanding data representation; it is an intelligent system and helps in making intelligent decisions [29]. In many facial analytical tasks, including face detection, age, gender prediction, and face recognition, deep learning-based systems demonstrate exceptionally high precision, leading many researchers to claim superhuman efficiency in face recognition. Recently, deep learning, specifically the convolutions neural network (CNN), has made excellent progress in the technology of face recognition [30].

The recent progress in deep CNNs has substantially improved state-of-the-art performance in face recognition, and there are a number of models available, such as AlexNet [30], ResNet [30], VGG [31], ArcFace [32], etc. In many computer vision applications, such as object detection and recognition and image synthesis, deep learning-based systems demonstrate extremely high accuracy [30][31]. Training CNNs for face recognition is complex, time-consuming and it is greatly advance for face

recognition performance. For instance, the accuracy of the Labeled Faces in the Wild (LFW) dataset has increased to 99.83 percent [32].

Recently, convolutional neural networks in deep learning have outperformed classical approaches. Deep learning algorithms on standard face recognition datasets achieve state-of-the-art results. VGGFace and VGGFace2 models are some examples of state-of-the-art models.

VGGFace: VGG Face is a dataset proposed for the training of deep models containing about 2,622 people with 2.6 million faces. Around one thousand samples, most of which are of high-quality frontal faces, describe each subject [33]. VGGFace's contribution is to demonstrate the creation of a very large training dataset needed to train modern-convolution-neural-network-based face recognition systems used to compete with the large datasets used by Facebook and Google to train models.

VGGFace2: VGGFace2 is an updated version of VGGFace that has been developed to mitigate the shortcomings of its predecessor. VGGFace2 comprises an average of 3.31 million photographs of 9,131 subjects, and 362.6 media descriptions of an individual. [33]. VGGFace2 is designed to cover a wide variety of poses, age, and ethnicity, and as much as possible to minimize label noise. [34].

An autoencoder is a kind of artificial neural network that is unsupervised and learns to copy its input into its output. It has an internal or hidden layer that defines a code that applied to define the input. The deep learning-based autoencoder has showed outstanding potential to significantly improve the performance of the physical layer [35]. An autoencoder aimed at learning a representation for a data set, typically, by training the network to avoid signal "noise" for dimensionality reduction. Autoencoders are composed of four main parts: - Encoder, bottleneck, decoder, and Reconstruction Loss. An encoder maps the input to the code and compresses an encoded representation of the input data. The bottleneck is the layer containing the representation of compressed input data and a decoder which maps the code to the first reconstruction of the input. Finally, Reconstruction Loss measures how nicely the decoder is achieving and how close the output is to the original input.

2.6 Challenges in Face Recognition

Recently, researchers focus more on unconstrained face recognition containing various poses, illuminations, expressions [29].

Pose Variations Challenges: Pose variation is a challenge in Face Recognition (FR). The literature review found that algorithms such as Pose Invariant Face Recognition (PIFR) degrade over 10 percent from frontal-frontal to frontal-profile face verification, including deep learning methods. The figure 2.4 below shows sample pose variation challenges [29].



Figure 2. 4 Pose Variation

Illumination changes: Lighting condition is one of the big factors for facial appearance changes and recognition performance degradation. The difference between the two images of the same person that is taken under varying illumination is visible than the two images that are taken in the same light conditions. The Figure 2.5 below shows an image of the same person that is taken in different light conditions [29].



Figure 2.5 Illumination Variation [29].

Challenges in Partial face images: The unavailability of the whole face is another challenge in a constrained environment. A Partial face image may occur when the person is occluded by objects such as faces of other individuals (masks, hats, sunglasses), when the image is captured in various poses without awareness, and when the person is partially out of the camera's field of view [29]. The figure 2.6 below shows partial face images.



Figure 2. 6 Partial Face Image Variation [29].

Facial Expression Challenges: Facial expression changes impose difficulties for face recognition too. Facial deformations with expression can change appearance. Researches have used a deep learning

method to address expression problems [29]. The figure 2.7 below shows images of the same person due to facial expression.



Figure 2. 7 Facial Expression Variation [29]

Mixed Variation: The other Deep learning methods are good at dealing with nonlinear variations in face images and making the extracted features more discriminative. Rather than focusing on one specific variation, there are a number of methods proposed, to address multiple mixed challenges, e.g. pose, illumination, expressions, and age. Mixed variations can be Pose + Illumination, Pose + Expression, Pose + Illumination + Expression, and multiple challenges [29].

2.7 Related Works

This section discusses relevant work on face recognition, face detection and forensics. Reviewing related to the subject area of face recognition has a vital role in choosing appropriate mechanisms to solve the problem and enhance the performance of the recognition system.

There are several methods proposed to solve pose variation and illumination problems such as works in [6], and [15]. In [15] the proposed method uses the idea of multi-band feature technique and multi-view training to overcome the illumination and pose variation problems respectively. The multiband feature technique functions to extracts the frequency sub-band that is most discriminant but invariant to illumination variation whereas several individual radial basis functions (RBF) neural networks are utilized to perform multi-view training.

The proposed system is then implemented on a database that has both pose and illumination variation to investigate its performance. The proposed system is then simulated on the YaleB database the results show that the proposed system has a better performance compared to the other existing systems. In [6] the proposed method uses a Discrete Curvelet Transform is used to extract features. Comparison with the traditional method shows a better recognition rate.

In 2017, Fabiola Becerra-Riera [4] proposed a new technique for merging present face recognition systems with the information acquired from facial marks in order to improve their performance. They

primarily use an algorithm to locate facial marks automatically, which are then represented using Histograms of Oriented Gradients (HoG), and they are matched taking into account their location within the face image. Extensive tests are performed to demonstrate the feasibility of the proposed algorithm for the identification of facial marks and to corroborate the benefits of using facial mark information on top of traditional facial recognition systems.

Anil K. Jain and Unsang Park [12] introduce the use of micro features, specially facial marks, to enhance face recognition and performance recovery and they use Facial marks in three ways:

- In an available face matcher, to add the features,
- Using facial mark based queries to allow quick retrieval from a large database, and
- To allow a partial face image or profile face image with markings to be matched or retrieved.

Pedro Tome and Ruben Vera-Rodriguez [36] A novel facial feature-based approach (hereinafter referred to as soft biometric facial features) was proposed, which can define the shape, orientation, and size of the facial features. The developed biometric framework has been designed to support Forensic examiners' job when conducting facial image morphological comparisons. The first phase is to extract facial landmarks. Then, to obtain continuous and distinct facial features, an image pre-processing stage specifically designed for each facial trait is completed. A forensic approach follows the proposed procedure and can therefore automatically provide an indicator of the shape, orientation, and size of the corresponding facial features. This can guide and support the judgment of forensic examiners.

This study also carries out an in-depth analysis of the individuality and discrimination power of such anthropometric-based facial features, since they are now usually used in many realistic cases without a scientific examination of their error rates.

In general, all the reviewed related works are done on face detection and recognition. Each research work contributes more or less to maximize the efficiency and accuracy of the algorithm but there are still gaps when a face is enrolled in the recognition software, usually, multiple angles are used and Illumination can be considered a complex problem in both indoor and outdoor face matching.

Deep learning, particularly deep convolutional neural networks, has become increasingly interested in face recognition [29]. Nowadays, there are a number of deep learning methods because due to an increase in the amount of data. One of the most common techniques in computer vision, such as image classification, object detection, face recognition, has become the Convolutional Neural Network (CNN).

2.8 Summary of Related Work

The following table illustrate previous works on face detection and recognition.

<i>Title</i>	<i>Dataset (Data collection)</i>	<i>Methodology used</i>	<i>Accuracy</i>	<i>Limitations</i>
Face Recognition Based on Deep Learning, Wang, Weihong, Jie Yang, and Dixin Zhou, 2014	-ORL face database -LFW face database	- Deep Convolutional Neural Network - Siamese CNN	- 91% on ORL face database - 81%on LFW face database	-Low recognition rate - does not consider PIE
Facial Soft Biometric Features for Forensic Face Recognition Pedro and Javier Ortega-Garcia, 2015	- ATVS Forensic database - MORPH database	- Geometrical landmarks are automatically extracted by geometry from the facial landmarks	- 100% for ATVS forensic database -75% for MORPH database	- does not consider pose variation It need retrain when new data is added
An Approach to Improve Recognition Rate from Occluded and Pose Variant Faces, Vijayalakshmi.A, 2015	-Youtube dataset	- Feature Extraction using Discrete Curvelet Transform - SVM is used for face verification	- The evaluation technique is not stated.	-it required longest time to recognize a face -It need retrain when new data is added
Component-based face recognition under transfer learning for forensic applications, Rupali Sandip,2018	-Faces94 standard face database	- Feature Extraction using CNN - Dimensional Reduction using FLDA - Using KNN classier	-Full face recognition rate of 98%	-it considers some facial features only -It need retrain when new data is added

Wavelet Transform based Feature Extraction for Face Recognition, Sangeeta Kakarwal and Ratnadeep Deshmukh, 2010	- FERET dataset	- Wavelet Transform based analysis method to extract the features	- 80%	-Low recognition rate. -It need retrain when new data is added
---	-----------------	---	-------	---

Table 2. 1 Related Works on Face Detection and Recognition

Chapter Three

3 Research Methodology

Different approaches used to achieve the general and specific objectives of the study and to solve the problem by answering the research questions.

The research methodology is a scientific way of understanding the problem domain and provides a solution in a scientific way. It also deals with different steps, along with the reasoning behind the research problem, to study the research problem.

In general, various steps are required to conduct this study. First, the researcher reviews different research papers related to this study for identifying research problems described in section 2. Second, this study collects data in order to generate a reliable result for face detection and recognition. This study's data collection includes querying reference databases with personalized search strings, followed by manual filtering of the results of the query, searched for papers published between 2010 and 2020. Then lastly, appropriate tools such as OpenCV and python were selected to develop the prototype and also constructs the model then evaluate the model, and finally, evaluate the performance of the system based on tests result.

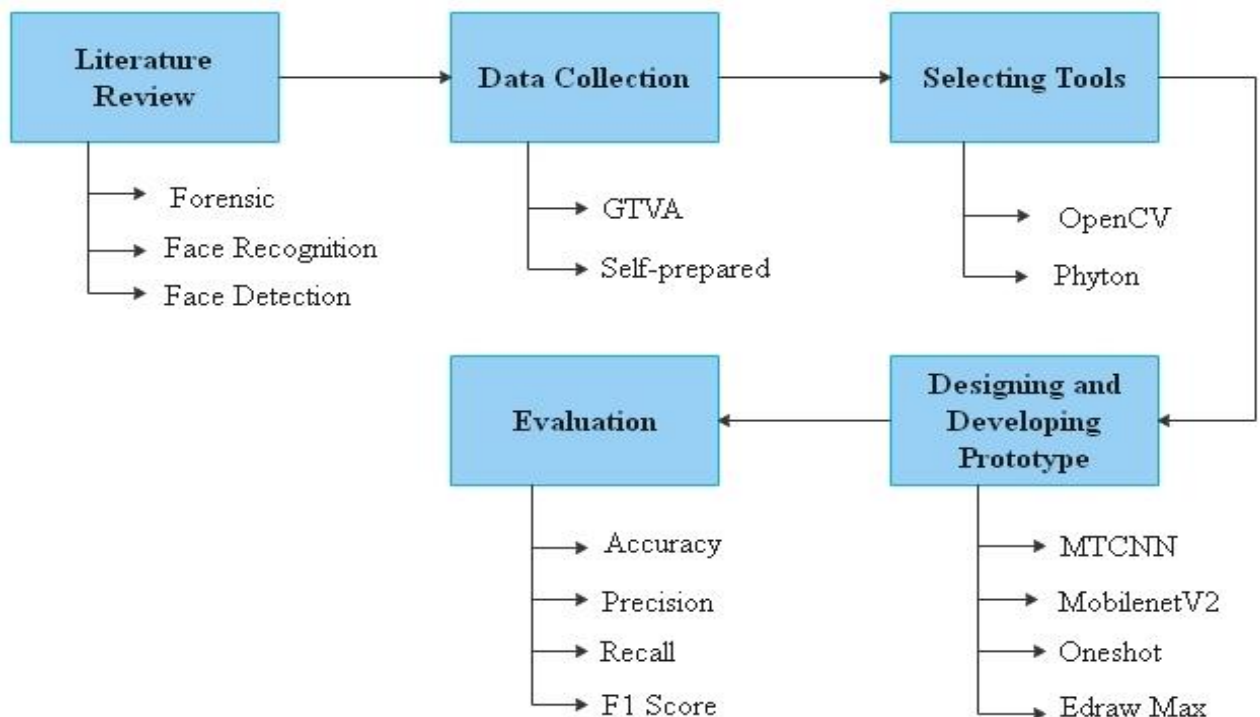


Figure 3. 1 Research Methodology

3.1 Literature Review

Reviewing related works is necessary to contribute their idea to describe what has been done and needs to be done in the area. Literature survey within the area of face detection and recognition has been reviewed from different sources including, conference papers, journal articles, Internet, and books. This supports the investigation of the fundamental principles of different approaches, methods, algorithms, and tools that would be best for this study. The Multi-task Cascaded Convolutional Neural Networks face detection method, One-shot learning, Siamese Network, Deep Convolutional autoencoder, and MobileNetv2 face recognition methods were part of the review. In addition, to understand how the OpenCV-Python works, various OpenCV and Python books were part of the review.

3.2 Research Design

The research design relates to the overall methodology you want to coherently and logically combine the different components of the analysis. Thereby, make sure you will effectively address the research problem and it is a framework that has been created to find answers to the research questions. Furthermore, a research design is specified as a plan that describes how, when, and where to gather and analyse data [16]. This supports us to carefully consider the research and to plan how they will approach the research. It also plays an essential role to integrate the various components of the study coherently and logically.

This study followed experimental research design and applied different methods starting from problem identification, define the objectives for a solution, Design and prototype development, testing and evaluating the performance and this study is all about enhancing the effectiveness, and increasing the accuracy of face recognition by using hybrid techniques.

3.2.1 Problem Identification

Identification of problems is the first and most significant step in the research process. This research begins by defining the problem of research clearly. This paper has defined a research problem by conducting a deep literature review on face detection, face recognition, and current challenges on face recognition. Based on the problem definition, we reduce the problem conceptually to taking its complexity for developing the face recognition model that can effectively provide a solution. This paper has emphasized in developing a face recognition model that can handle pose-variation, illumination, and facial expression.

3.2.2 Defining Objectives of the Study

The researcher assumed the objectives of a solution and knowledge of what is possible with face detection and recognition, based on the knowledge of the state of problems. This paper has described both general and specific objective depending on how our proposed algorithm expected to support solutions to the problems.

3.2.3 Designing and Developing Prototype

In order to develop a solution for the listed research questions, this paper is used state of the art approaches for face and facial landmarks detection, and face recognition,. According to the literature review, Multi-Task cascaded convolutional Neural Network(MTCNN) is the state of the art face detection algorithm where as MobileNetV2 is a state-of the art feature extraction algorithm with One-Shot Siamese Neural Network for face recognition. Edraw Max is an extremely efficient type of software design for all-in-one diagramming that can operate on several platforms is used.

In addition to selecting face detection and face recognition algorithms, we also select tools and packages for developing our proposed solution to the stated problems. This paper has selected OpenCV python, keras, and Tensorflow tools for ce recognition model.

3.3 Data Collection

In order to build or train and reliably test the proposed face recognition model, sizable databases of face images are needed. Many face databases to be used for uncommercial purposes are accessible on the Internet. In this study, all data's required for the experiment are collected from well-known public image databases available on the Internet and self-prepared database. These databases are prepared under various pose, illumination and facial expression.

3.3.1 GTAV Face Database

The main purpose of the GTAV Face Database [37] was to assess the robustness of face recognition algorithms on facial expression, strong position, and variations in illumination. In this study, experiments are conducted on GTAV datasets. It includes 1,204 images a total of 44 different peoples with 30-51 pictures per person, which is about 27 images per class on average, which correspond to different pose views (0° , $\pm 30^\circ$, $\pm 45^\circ$, and $\pm 60^\circ$) under three various illuminations (an almost frontal mid-strong light source, natural light or environment and finally a strong light source from an angle of 45° .) In addition, with various occlusions and facial expression variations, a minimum of five more additional frontal view images are included. The images are with resolutions of 240×320 and they are in BMP format [37]. The dataset contains images of 44 classes. The dataset contains multiple images of a person face from pose variation, illumination and facial expression. In these figures below present some example of the database.



Figure 3. 2 Example of a Person ID on the Database (Different Pose Views)



Figure 3. 3 Example of a Person ID on the Database (Different Illumination)

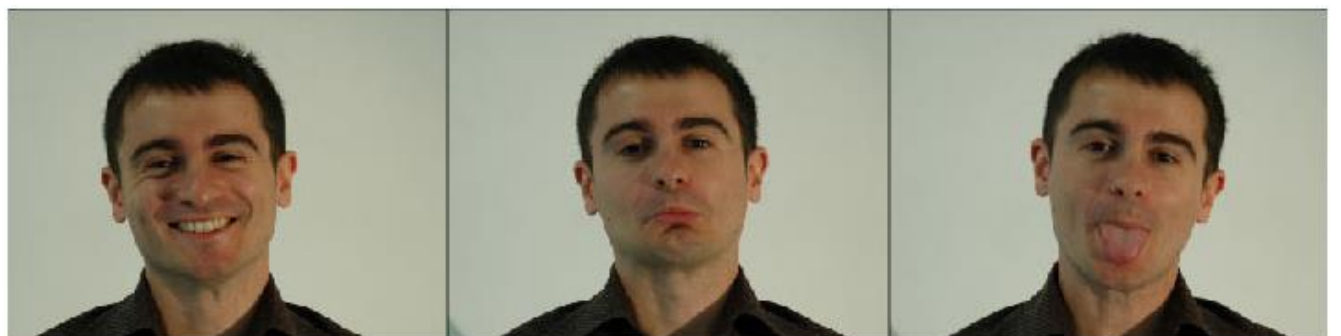


Figure 3. 4 Example of a Person ID on the Database (Different Facial Expression)

3.3.2 Self-Prepared Database

Testing the performance of the algorithm using a self-prepared dataset is a necessary part of the study. To test the proposed face recognition algorithm additional we have prepare our own face database. The database is Contains of 80 coloured images in Jpg format of 20 individuals with various illumination, pose, and facial expression. The images are collected by using Samsung Galaxy A505.



Figure 3. 5 Example of Persons on the Self-Prepared Database

3.4 Multi-task Cascaded Convolutional Neural Networks (MTCNN)

Multi-task Cascaded Convolutional Neural Networks (MTCNN) is an algorithm that has been developed for both face recognition and face alignment as a solution. The process consists of three stages of convolutional networks that can identify faces and landmarks such as eyes, nose, and mouth [38].

Deep learning algorithms have recently been used to detect faces in images and videos using Tensorflow. There are many face detection works that follow Haar cascade network for detecting frontal faces. However, Haar cascade networks do not work with lightweight variations. This study uses frontal face detectors based on MTCNN that use a cascade of neural networks. Using multi-task learning, MTCNN combines all tasks (recognition and alignment). It implements a shallow CNN at the beginning to create candidate windows quickly. In the second step, the proposed candidate windows are refined through another complex Convolutional Neural Network. Finally, to further refine the result and output facial landmark locations, it uses a third CNN, more complex than the others [38].

MTCNN works in three stages: -

The first step taking an image and resize it to different scales to create an image pyramid. This step is an input to the three-cascaded network. The first stage is a called P-Net, which produces candidate windows by a shallow convolutional neural network.

The second stage is a deeper CNN called R-Net is to reject as many non-face windows as possible. The last stage is O-Net is a complex network to further refine the output of R-Net [38].

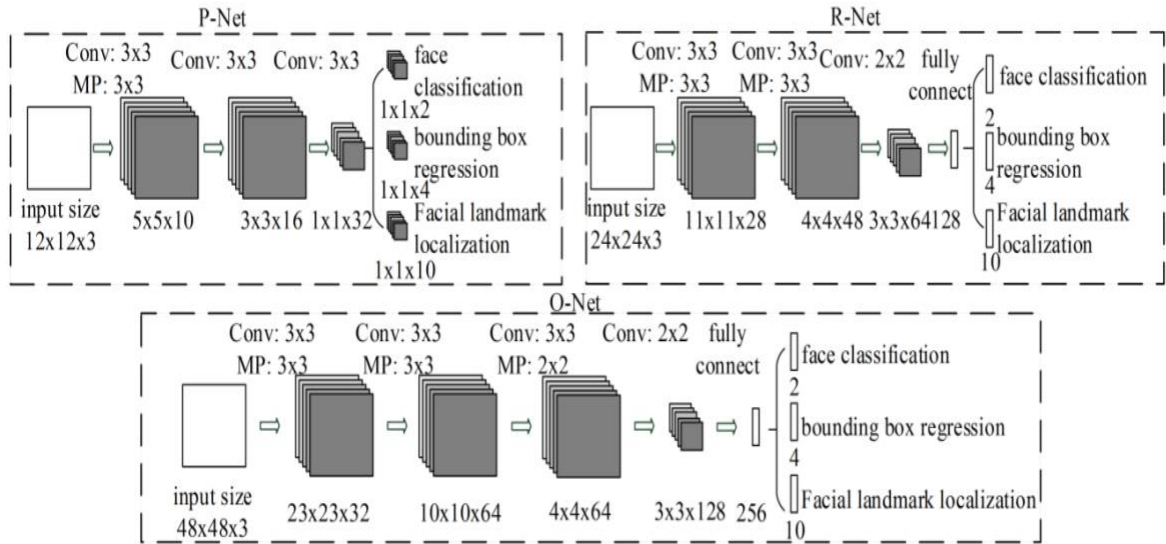


Figure 3. 6 The Architectures of P-Net, R-Net, and O-Net [38]

3.5 Feature Extraction for Face Recognition Selection

Deep neural networks and Deep Learning are strong and current algorithms. In addition, a lot of their achievement lays within the careful design of the Neural Network architecture. Various states of art Deep CNN based architectures have been proposed over the last decades for the classification of images. The selection criteria include the accuracy, speed, and number of data.

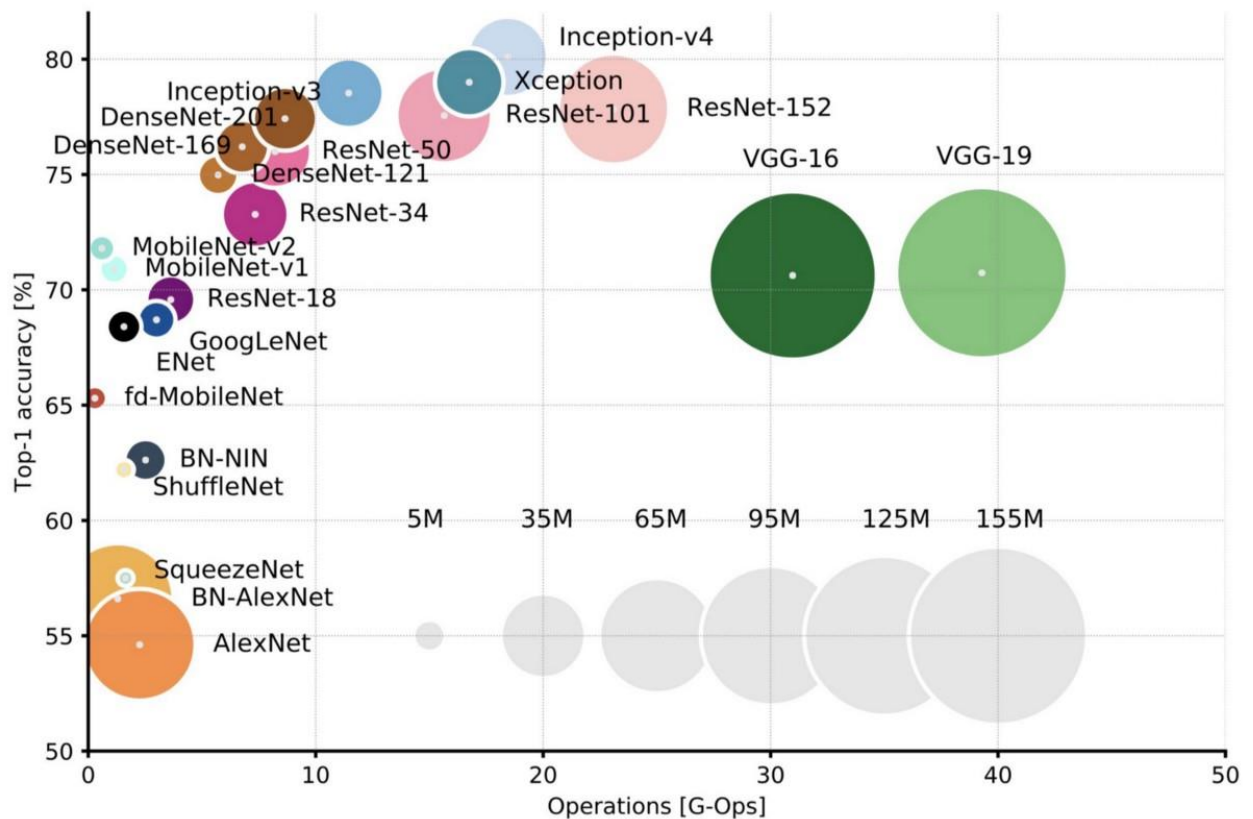


Figure 3. 7 Benchmark Analysis of Deep Neural Network Architectures[39]

MobileNetV2 has less than 5 million parameters where others architectures like xception has more than 5 million parameters, ResNet has more than 35 million parameters and VGG has more than 155 million parameters. MobileNetV2 architecture uses separable convolutions to reduce the number of parameters. It has less number of parameters and operations compared to most architectures, which makes it able to run fast on different devices that have less computing capabilities. Based on the above observation for feature extraction this study used MobileNetV2.

3.6 MobileNet Convolutional Neural Network

Convolutional neural networks have the capability to self-adapting, self-learning, and self-organizing; so, it can automatically extract features by using the prior knowledge of the recognized categories and avoid the complicated process of feature extraction in traditional image classification techniques. Deep convolutional neural network (CNN) has achieved significant success in the field of computer vision, such as image classification, target tracking, target detection, and semantic image segmentation.

Krizhevsky et al. Won the ImageNet Visual Recognition challenge in 2012 with AlexNet. AlexNet is a Deep Convolutional Neural Network with 60 million parameters and eight layers. In addition, great success has also been achieved by VGG with 16 layers, Google Net with initiation as a basic structure, and ResNet with residual blocks that can alleviate the gradient disappearance problem. However, the huge number of parameters, heavy computational load, and a large size of memory access lead to a huge power consumption which makes them apply for a portable mobile device with a limited number of hardware resources [33].

MobileNet has a higher classification accuracy because it is a lightweight deep neural network due to having fewer parameters [33] [40]. To further decrease the number of network parameters and increase the accuracy of the classification, the dense blocks introduced in DenseNets are implemented in MobileNet [34] [40].

There are two Dense-MobileNet models these are Dense1-Mobile Net and Dense2-Mobile Net. Experiments show that Dense2-Mobile Net can achieve higher recognition accuracy than Mobile Net and computational cost, but only with fewer parameters [33]. MobileNet is a simplified architecture that develops lightweight deep convolutional neural networks using depth-wise separable convolutions and provides an effective model for mobile and embedded vision applications [40]. The Figure 3.8 below shows the architecture of MobileNet.

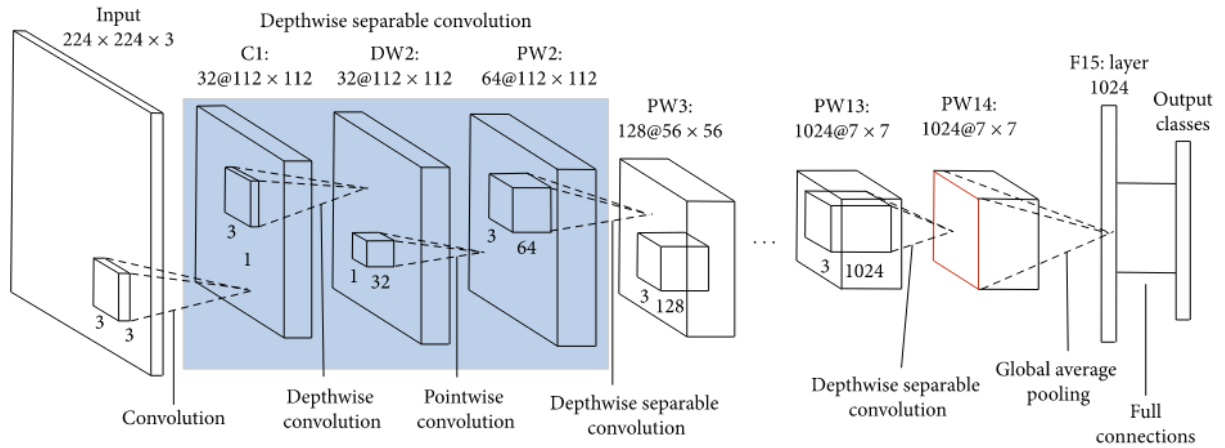


Figure 3. 8 Architecture of MobileNet [33]

3.7 One-Shot Learning

This is an object categorization problem and typically, machine-learning algorithms need training on lots of or thousands of samples that result in a big dataset. However, One-shot learning purposes to learn information from one or a few samples. In machine learning, learning from a few samples remains a key challenge. One-shot learning is a classification task in which for each class worked to prepare a model, a very limited number of examples or a single example is presented that must, in turn, enable predictions about several unknown examples in the future. One-shot learning can be performed using Siamese Network [41].

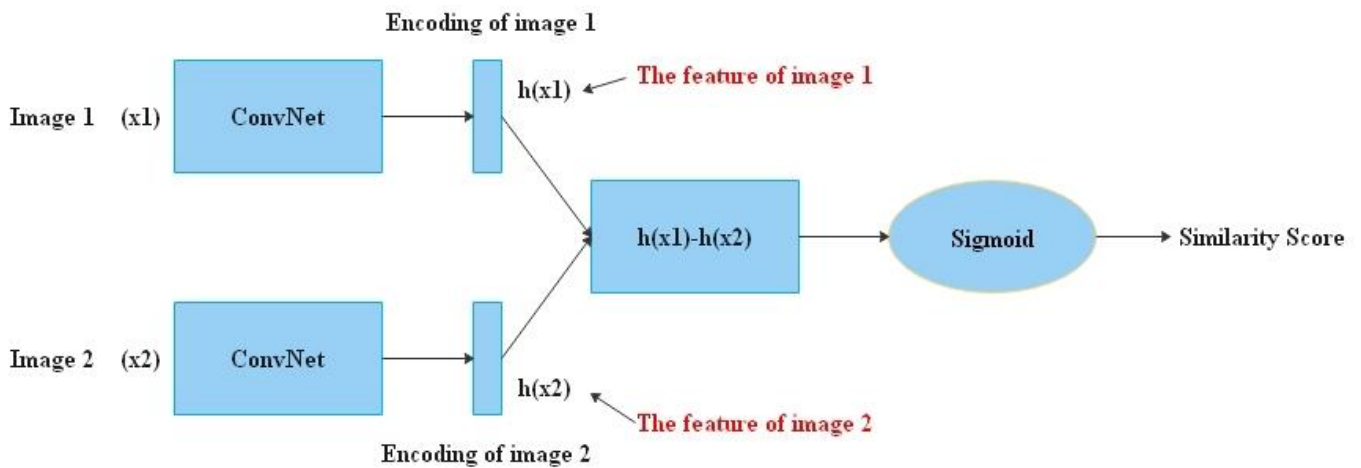


Figure 3. 9 Siamese Network Architecture

A Siamese network consists of two identical neural networks, every taking one between the two input pictures. A contrasting loss function that calculates the similarity between the two given images then fed to the last layers of the two networks. Every image within the image try fed to at least one of those networks.

3.8 Development Tools and Packages

3.8.1 OpenCV-Python

OpenCV stands for Open Source Computer Vision Library, an open-source image and video analysis software library for computer vision and machine learning that was originally introduced by Intel more than a decade ago [42]. Several programmers have since contributed to the most recent advances in libraries. OpenCV was designed to present a common infrastructure for applications for computer vision and to speed up the value of machine understanding in commercial products. Being a BSD-licensed product [42], OpenCV makes it simple for businesses to employ and modify the code.

The OpenCV has over 2500 optimized algorithm libraries; it incorporates a comprehensive collection of algorithms for computer vision and machine learning, both classic and state-of-the-art. These algorithms are also used to detect and recognize faces, classify people's activity in videos, recognize objects, monitor moving objects, track camera movements, extract 3D object models, stitch images together to create a high-resolution image of an entire scene, search similar images from a database of images, displace red eyes from flash images, identify scenery following eye movements, and set markers for augmented reality overlay, etc.

OpenCV has over 47 thousand individuals in the user community and the supposed number of downloads exceeding 14 million [42]. The library used extensively in research groups, companies, and governmental bodies. OpenCV has Python, Java interfaces, and C++/C with support for Mac, Linux, Windows, iOS, and Android.

Python is a general-purpose programming language that, essentially because of its simplicity and code readability, became very popular in a short time. It makes it possible for the programmer to communicate ideas in fewer lines of code without decreasing any readability. Python decreases the cost of program maintenance, simple and easy to learn syntax emphasizes readability. Python sustains packages and modules that enable the modularity of programs and the reuse of code. Compared to other programming languages such as Java, C++, and Ruby, the fast development time.

- There are plenty of libraries in python that makes our task easier.
- It has interactive (has a terminal for debugging and testing) and portable (runs on a variety of hardware platforms).
- Vital feature of Python is that it can be simply extended with C/C++. This feature allows us to write C/C++ computationally intensive codes and create a Python wrapper for it in order to use these wrappers as Python modules. This gives us two benefits: first, our code is as quick as the original C/C++ code, and second, coding in Python is very easy to do and the support of Numpy makes the work simpler.

3.8.2 TensorFlow

TensorFlow is an end-to-end open-source platform that has flexible tools and resources for machine learning. Many states of the art NLP applications are developed using TensorFlow as a backend. Popular organizations like Google, Intel, and others also use TensorFlow to develop systems [7].

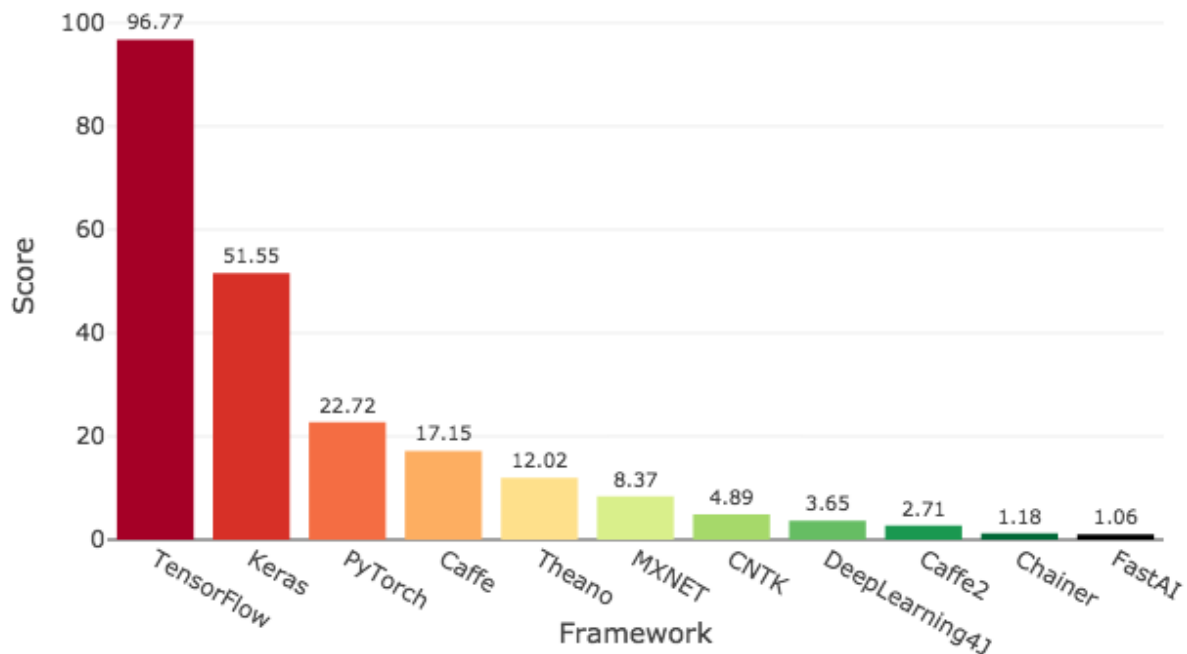


Figure 3. 10 Deep Learning Framework Power Score 2018 [43].

3.8.3 Keras

Keras is a library for developing deep neural networks in python that can run using TensorFlow or Theano as a back-end to pre-process data, to create, evaluate, optimize, fit and test a model [7]. For the development and shipping of machine learning solutions with high iteration speed, Keras gives basic abstractions and building blocks [44].

3.8.4 Numpy

Numpy is a library for python that can solve scientific computation easily, for numerical operations, is an extremely optimized library [7]. It gives a MATLAB-style syntax. All structures of the OpenCV array are translated to-and-from Numpy arrays. Therefore, you can merge whatever operations you can do in Numpy with OpenCV. In addition, some other libraries can be used with this, such as SciPy, Matplotlib that supports Numpy. Therefore, OpenCV-Python is a suitable tool for quick prototyping of problems with computer vision.

Pandas: Added on top of the Python programming language, Pandas is an efficient, fast, versatile, and easier to be using open-source data analysis and manipulation tool [7].

3.9 Evaluation Metrics

In order to measure the performance of a model, there are many evaluation metrics such as accuracy, precision, recall and F score [16].

These measures are based on

True Positive (TP): The number of instances that are positive and correctly predicted as positive.

False Positive (FP): The number of instances that are negative but incorrectly predicted as positive.

False Negative (FN): The number of instances that are positive but incorrectly predicted as negative.

True Negative (TN): The number instances that are negative and truly predicted as negative

Accuracy (A): refers to how close a measurement is to the true or accepted value or how much accurately the model learns to classify the data [16].

$$A = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

Recall (R): is calculated as the fraction of true positives from the sum of True positives and False Negatives [16].

$$R = \frac{TP}{TP+FN} \quad (3.2)$$

Precision (P): is determined from the sum of the related groups as the fraction of true positives (TP), i.e. the sum of true positives and false positives [16].

$$P = \frac{TP}{TP+FP} \quad (3.3)$$

F score (F): If we need a compromise between accuracy and recall, it is a simpler and better-known metric to use. It interpreted as the harmonic mean of precision and Recall [16].

$$Fscore = \frac{2*Recall*Precision}{Recall+precision} \quad (3.4)$$

Chapter Four

4 Proposed Component Based Face Recognition Technique

4.1 Overview of the Proposed Technique

In this section, the proposed method for face detection and face recognition are described in detail. In order to accomplish the general and specific objectives of the study. To solve the gaps stated in the literature review, a hybrid one-shot face recognition approach has been proposed. This paper presented two main procedures, the first procedure is face detection, In the Hybrid Face Recognition Technique face detection is used to identify either it is face or not. This study is used Multi-task Cascade Convolution Neural Network (MTCNN) to automatically detect the faces and in the same time it detects face landmark [45].

Face detection using MTCNN is goes through three stages to output a proper face detection. The candidate windows and their bounding box regression vectors are obtained in the first stage, combining highly overlapping candidates [45]. The second stage provides another CNN with those candidates, which declines a large number of false candidates. The third stage is similar to the second one, but it also outputs five facial features points, two eye centre points, nasal tip point and two corners of the mouth points [46].

In general, various steps are required to conduct this study. The researcher followed the following stages. First, the pre-processing stage, in which we resized each image to a suitable size for each CNN model and converted any grey images to RGB images. From the pre-processed image, the face is identified utilizing Multi-task Cascade Convolution Neural Network (MTCNN). In addition, the third stage is Applying the pre-trained CNN for extracting features. Image features of contrast, energy, correlation, cluster prominence, maximal probability, entropy, homogeneity, cluster shade, local homogeneity, a total of squares or variance, dissimilarity, autocorrelation and inverse variant moment are extorted, once the face detection is accomplished in images.

The second procedure is face recognition; this study is used MobileNetV2 and Siamese Neural Network to recognize face. Face Recognition technology consists of four stages, which include face detection, alignment, representation (facial feature extraction), and classification or verification [46]. Figure 4.1 below shows recent general approaches the researcher followed for face recognition.

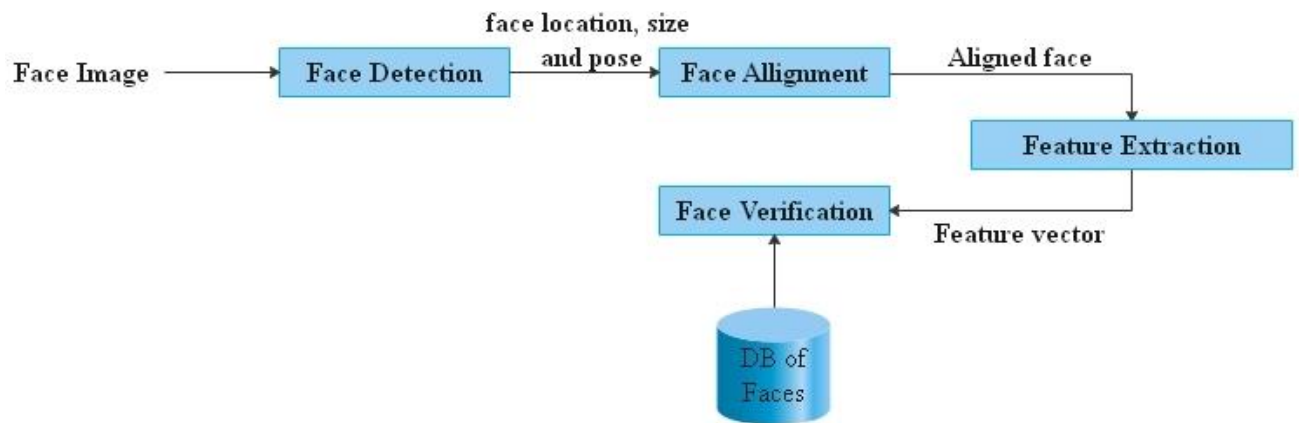


Figure 4. 1 General Face Recognition Process Flow [46]

Taking the general approach as reference, we have developed our proposed approach that can answer the research questions. In our proposed approach, additionally this study use facial landmarks for better performance. Figure 4.2 below shows the proposed approach for face recognition that can handle pose variation, illumination and expression.

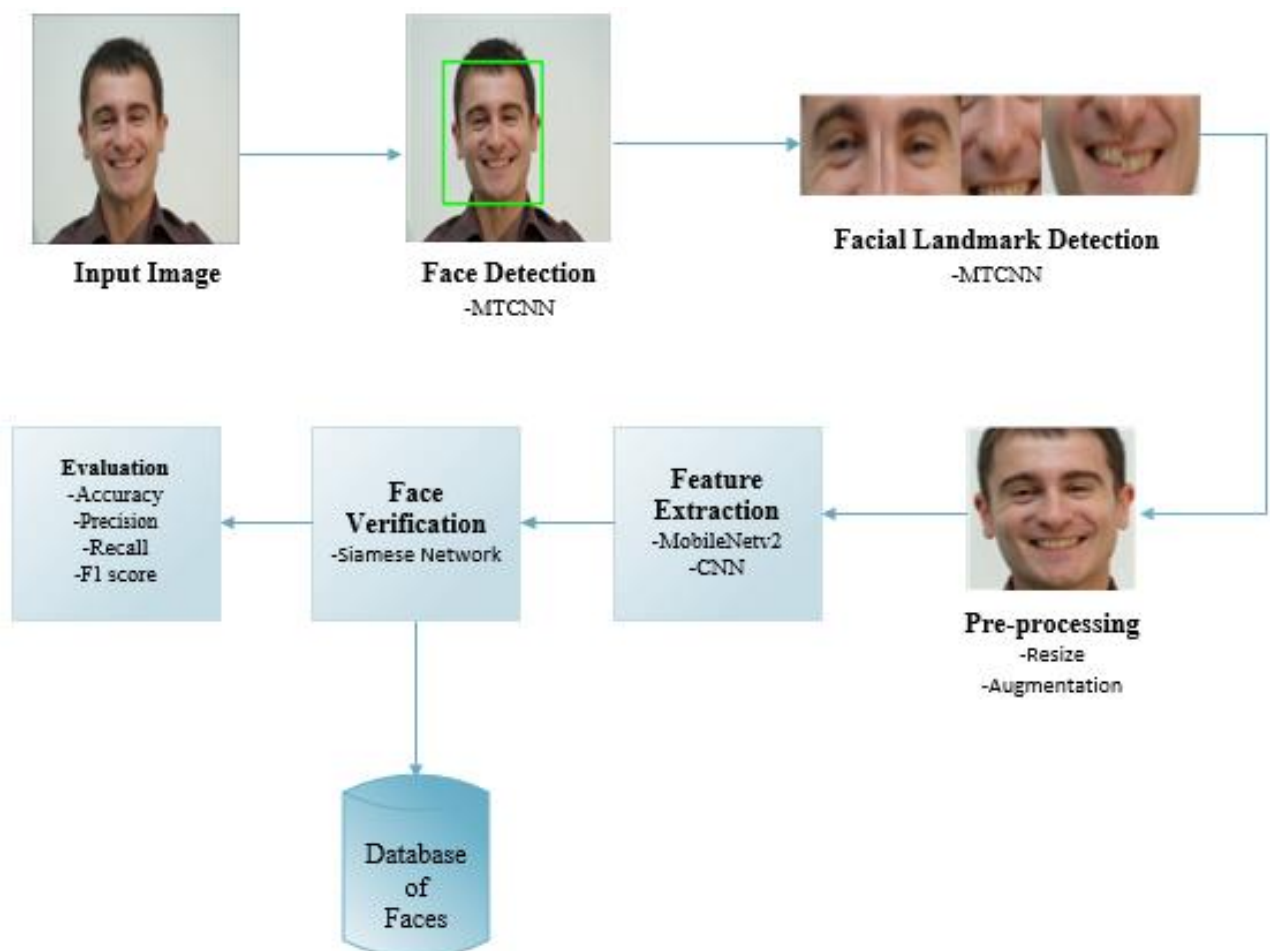


Figure 4. 2 Proposed Approach for Face Recognition

Each steps of our proposed approach discussed as below

4.2 Pre-Processing of Image for Face Detection

Preprocessing the image is of vital importance on the FR. The main purpose of applying image preprocessing is necessary to remove information that is not relevant for feature extraction. Many researchers have presented several research works to conduct preprocessing before going into face recognition phase. As described in [16] the main purpose of image pre-processing is to produce a clear image data for the processing phase. Some image preprocessing includes Image transformation, image resizing, histogram equalization and Geometric normalization. Figure 4.3 below shows the steps followed to prepare our face database for face detection.

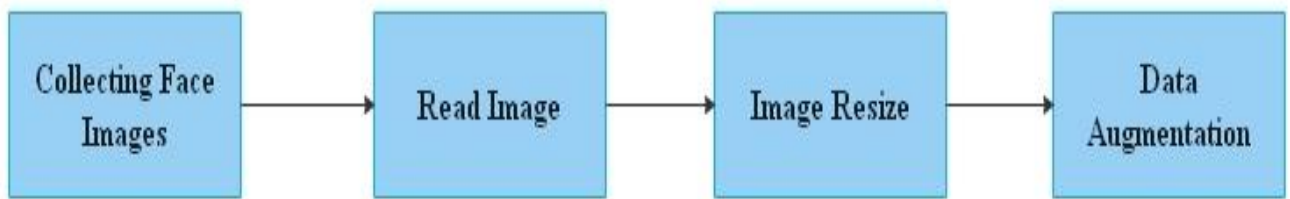


Figure 4. 3 Pre-processing steps for Face Recognition

4.2.1 Collecting Face Image (Get face image datasets)

Prepare images of persons that have illumination, pose variation and expression. The face databases in our proposed system are GTAV Face Database and Self-prepared Database used.

4.2.1.1 Read Image

Reading human face from a certain directory is the first and fundamental process in face recognition. A computer converts a human face image into an array of pixels that a value ranges from 0-255. Color images are a combination of three primary channels called RGB these are Red, Green, and Blue. A color image read using OpenCV imread module.

4.2.1.2 Image Resize

The next task in image recognition is resizing the image. Images have different size in terms of width and height. Image resizing to a common size can be done using OpenCV library. The OpenCV library command that used to resize an image is `cv2.resize (src, dsize [, dst [, fx [, fy [, interpolation]]])`

Where

src → stands for the source, converting the original image into Numpy array.

Dsize→ the required size of the output image is basically the desired one. The nature of the size is a tuple.

fx → the scaling factor along the X-axis or Horizontal axis is used for

fy→ is also used for the Y-axis or Vertical axis scale factor.

Interpolation → is a method based on the neighbouring pixels and the scale at which the image is resized to determine the pixel value. The interpolation options can be INTER_NEAREST, INTER_LINEAR, INTER_AREA, INTER_CUBIC, and INTER_LANCZOS4.

4.2.1.3 Data Augmentation

Image data augmentation is a technique used to create a new training dataset from existing training data [47]. Training Deep learning models with more data can result in more skillful models than training with small data. Augmentation techniques create variations of the images that can step-up the capacity of the model to generalize what they have learned to new images.

It is important to carefully choose the selection of the particular data augmentation technique used for a training dataset. The choice of Augmentation technique depends on many factors such as; I) variations of the training set images should be likely to be seen by the mode, ii) depend on the context of the training dataset, and iii) problem domain. This study used three types of augmentation techniques such as -

- **Horizontal Flip Augmentation:** An image flip is a means of reversing the rows or columns of pixels vertically or horizontally [47][48].
- **Random Brightness Augmentation:** By brightening images, darkening images, or combining these methods, the brightness of an image can be augmented. The image is darkened by values less than 1.0, the image is brightened by values greater than 1.0 where the value equal to 1.0 has no effect on brightness [48].
- **Gaussian Bluer augmentation:** Blurring an image requires taking and averaging neighboring pixels [47]. When different amount of blur is implemented, it is determining how many neighbouring pixels included. This spread from a single pixel is calculated as a standard deviation in both horizontal(X) and vertical(Y) directions. The greater the standard deviation, the more the picture gets blurred [48].

4.3 Face and Facial Landmark Detection

Face detection is a non-trivial computer vision problem for identifying and localizing faces in images or videos. There are several face recognition techniques, according to the literature. Using the OpenCV library, face detection can be carried out using the classical feature-based cascade classifier. Nowadays, on face detection datasets, deep learning techniques have reached state-of-the-art results. Face detection depends on deep learning can be performed through the MTCNN library applying a Multi-Task Cascade CNN.

Face detection using MTCNN is goes through three stages to output a proper face detection. Candidate windows and their bounding box regression vectors are acquired in the first stage, combining highly overlapping candidates [45]. The second stage provides another CNN with those candidates, which declines a large number of false candidates. The third stage is similar to the second one, but it also outputs five facial features points, two eye center points, nasal tip point and two corners of the mouth points.

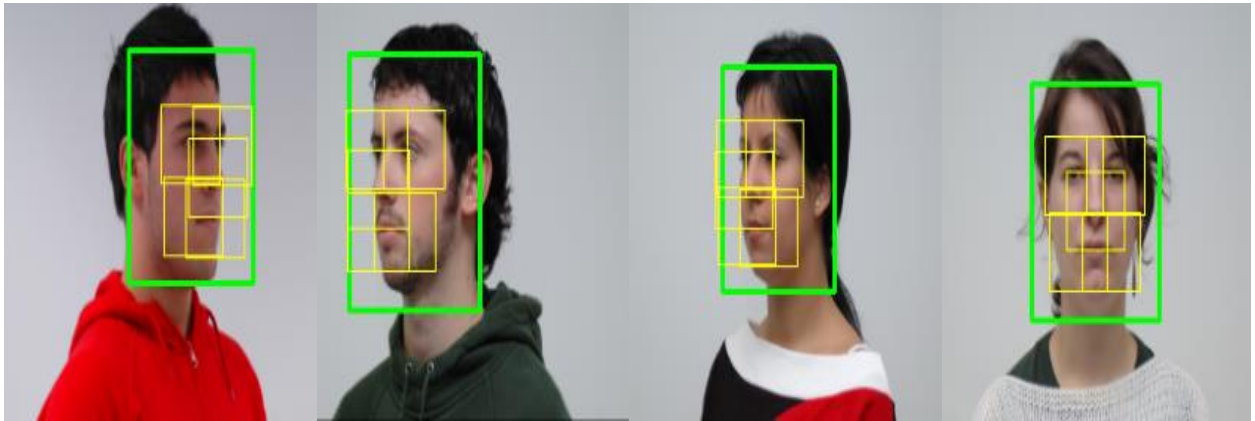


Figure 4. 4 Example of Face and Landmark Detection

The method consists of three phases of convolutional networks that can recognize faces and locations of landmarks such as the left eye, right eye, nose, left mouth corner, and right mouth corner. The Three Stages of MTCNN:

Stage 1: The Proposal Network (P-Net)

A Fully Convolutional Network (FCN) could be this first stage. This stage is chargeable for proposing different regions of faces to following stage [45]. For inducing candidate windows and their bounding box regression vectors, this Proposal Network is used.

Bounding box regression, a popular tool for predicting the localization of boxes when the face of some pre-defined class is identified by the objective. After obtaining the bounding box vectors, some refinement is completed to mix overlapping regions. All candidate windows are the ultimate output of this stage after refinement to reduce the volume of candidates [45].

Stage 2: The Refine Network (R-Net)

This network is responsible for refining the proposed bounding boxes from predicted from P-Net. This network is not fully convolutional network it is a CNN.

The R-Net outputs if the input is a face or not, a four-element vector which is the bounding box for the face, and a ten-element vector for facial landmark localization [45].

Stage 3: The Output Network (O-Net)

Before taking R-Net's output as input for final prediction, O-Net uses both P-Net & R-Net, and this Output Network aims to explain the face in additional detail and output the five facial landmarks' right mouth corner, left mouth corner, nose, left eye, and right eye. Therefore, the task of the MTCNN Network is to generate three things: face or non-face classification, bounding box regression, and localization of facial landmarks [45].

4.3.1 Face Alignment

Face Alignment is a process of normalizing a face to be consistent with a database such as geometry and photo metrics [49]. The purpose of this alignment component is to generate from the input image a frontal face that can include faces from different poses and angles. There is no out of the box function for face alignment but use Face cascade, Eye cascade, and some trigonometry. This has proposed to use MTCNN for face alignment process [45][49].

4.4 Feature Extraction (Feature Representation)

The introduction of Convolutional Neural Networks (CNNs) have dramatically improved the state-of-the-art performances of many computer vision tasks including face recognition [50]. Deep CNNs have recently played a significant role in face recognition research by either changing the objective functions or redesigning the architectures of the network [51].

Well trained CNNs are capable of handling pose, occlusion, and light variations of faces considerably well [50]. In recent years, active research has been interested in the tuning of deep neural architectures that balance accuracy and performance [52]. From such numerous variant designs some of them are Residual Convolution Neural Network (ResNet), LeNet, AlexNet, VGGNet-16, and GoogleNet / Inception. Nowadays, there is a lot of progress in algorithmic architecture that included hyper parameter optimization. MobileNet is a light weight architecture with fewer parameters [33].

4.4.1 Feature Extraction (Feature Representation) using MobileNetv2

MobileNetV2 is a new Mobile Net architecture that enhances mobile models' state-of-the-art efficiency on various responsibilities and benchmarks as well as over a variety of different model sizes. [52]. This paper has proposed to use MobileNetV2 due to fewer parameters and higher classification accuracy. MobileNetV2 uses $k = 3$ (3×3 depth wise separable convolutions) so the computational cost is 8 to 9 times smaller than that of standard convolutions at only a small reduction in accuracy [52].

The feature extractor is constructed using MobileNetv2 and CNN architecture. The model has two inputs, the face image and facial component. The face image is given to the modified MobileNetV2 architecture and the component is given to the CNN architecture created from scratch. The last three

layers of the MobileNetV2 removed since the model trained on 1000 classes and our model has 44 classes. The feature from the modified MobileNetV2 and component from CNN architecture are concatenated and fed to another CNN architecture. A classification layer with 44 classes added in order to evaluate the model.

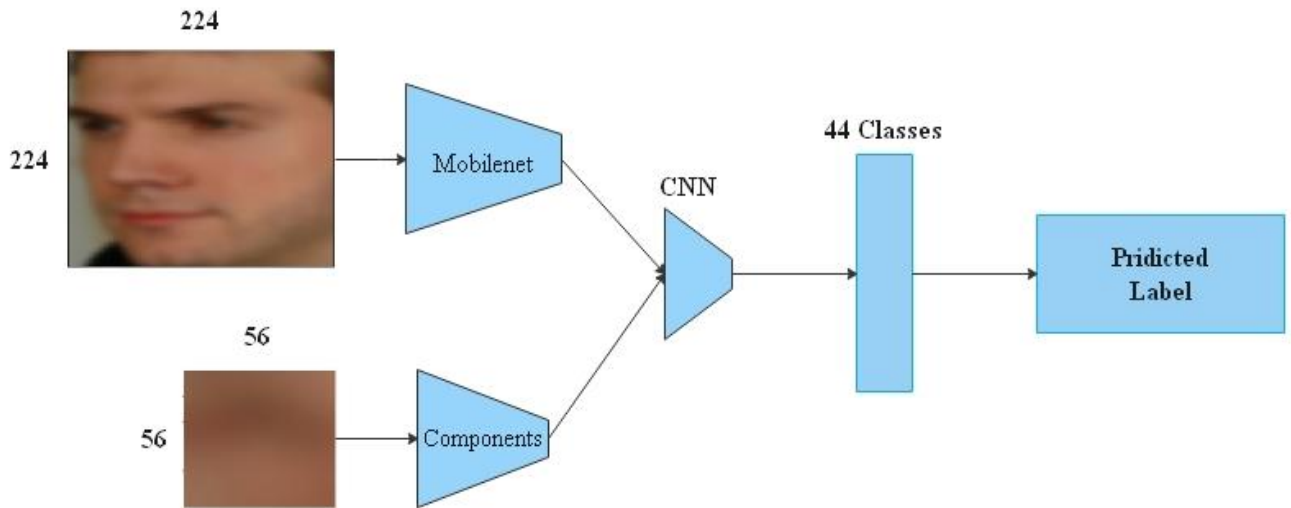


Figure 4. 5 Feature Extractor using Mobile Net

The idea behind creating this model is that, the same model used to construct the One-shot model. It will learn how to represent the face image and facial component.

4.5 One-Shot Model

One instance of an object class is introduced to the algorithm in the case of One-shot learning. Face recognition tasks give instances of one-shot learning. A model may only have one or a number of samples of the face of a given person in the case of face recognition and must correctly recognize the person from new images with changes in illumination, pose variation, facial expression, and more.

In the case of face verification, a model might at most have one example of a person's face image on record and should properly verify new images of that person, maybe daily. As a result, face recognition is a common example of One-shot Learning. When a deep learning model is adjusted for One-shot learning, it takes two pictures and returns a price that shows the similarity between the two pictures. If the images contain identical objects or identical faces, the neural network returns a worth that is smaller than a particular threshold and if they are not identical objects, it will be above the edge.

A Siamese Network a special type of Neural Network, and it is one of the best and most popularly used one-shot learning algorithms. A Siamese neural network consists of dual networks that settle

for distinct inputs. The objective of our network is to learn whether two faces are similar or dissimilar. The feature extractor constructed in the above section is used as the two identical networks sharing the same parameter. The output of the two identical networks is given to the Euclidean function in order calculate the similarity matrix and the passed to the sigmoid activation layer. The model output ranges between 0 and 1.

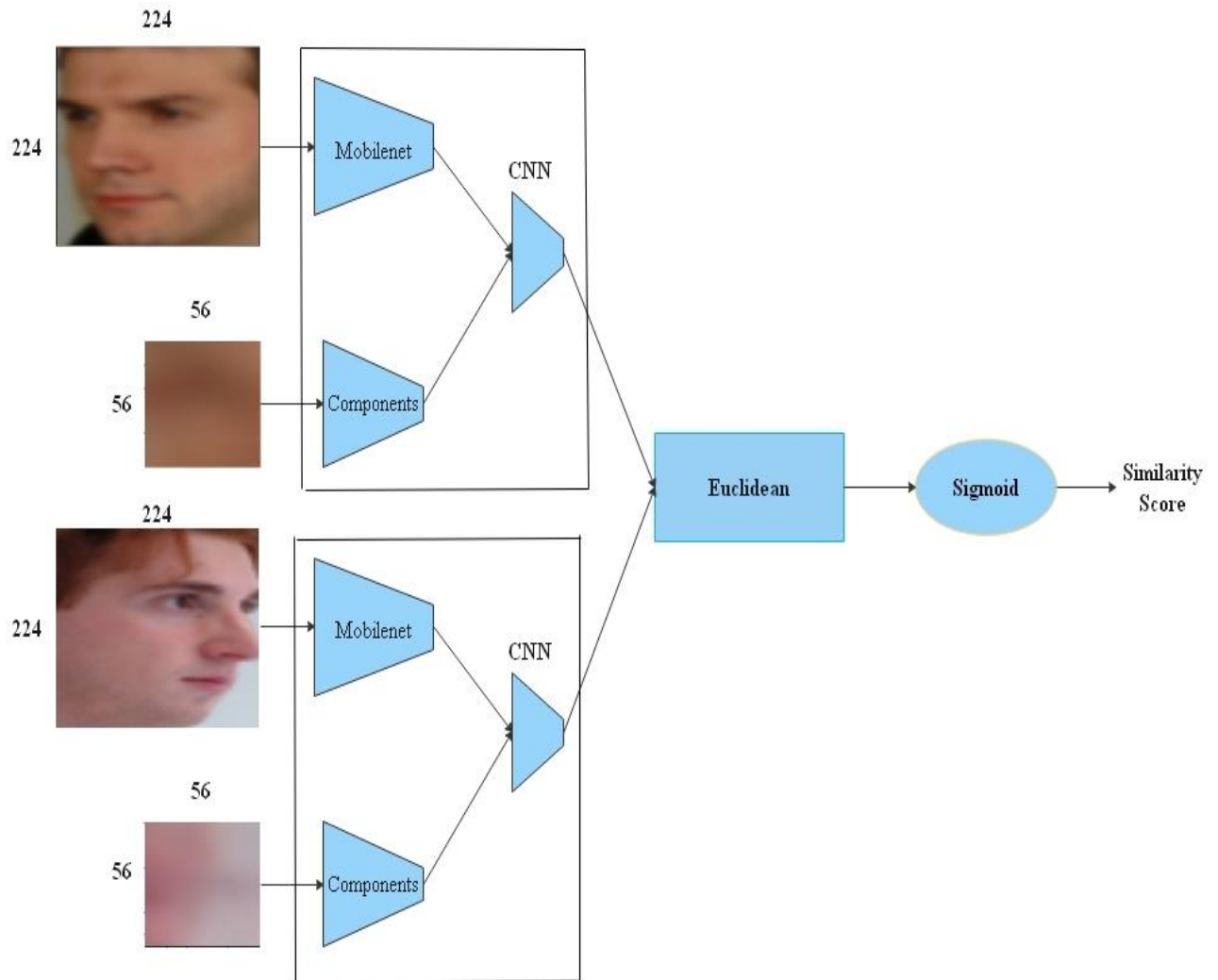


Figure 4. 6 The Proposed One-shot Model Architecture

The similarity score is used to decide whether the images are similar or not.

4.6 Evaluation

The result will be analysed to define the performance of the face recognition model on a test data set. The dataset is split into training validation and testing set. The proposed algorithm is evaluated using the test set. For evaluating the model's results or performance, a confusion matrix is applied. From the confusion matrix the model's Accuracy, Precision, Recall, and F1 score is calculated.

4.7 Database

In this study, the researcher used data required for the experiment are collected from well-known public image databases available on the internet called GTAV and Self-Prepared database is used. These databases are prepared under various illumination, pose variation, and facial expression.

Chapter Five

5 Implementation of the Proposed Face Recognition Technique

5.1 Overview

This chapter presents the implementation detail of the proposed One-shot based face recognition architecture. The implementation of the face detection, face component or facial landmarks detection and recognition architecture are discussed step by step in the following subsections.

5.2 Experiment and Development Setup

5.2.1 Working Environment

- Laptop: Lenovo computer used for Face Detection and Recognition development and testing. Table 5.1 shows hardware and software specification of the machine

Manufacturer	LENOVO
Model	Lenovo G40-60
Processor	Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz,2601MHz,2 cores,4 logical processor(s)
Memory	6GB
Operating System	Windows 10
System Type	64-bit Operating System, x64-based Processor

Table 5. 1 Hardware and Software Specification of the Machine

- Jupyter Notebook
 - Jupyter Notebook used to write code, with python interpreter 3.7 on the laptop computer.
- Google Colab
 - Google Colab used for training and testing the deep neural network with GPU and TPU support. The face recognition CNN architecture trained and tested in google Colab with GPU for faster training.

5.3 Dataset Preparation and Description

GTAV Face Database and Self-prepared Face Database created for testing the robustness of a face recognition algorithm, so it contains different poses, facial expression and illuminations variation. In order to train the proposed face recognition algorithm the face in these images detected, resize, and passed through augmentation process, since more data is required to train deep learning architecture. The following Figure shows sample images from GTAV face database after face detection.

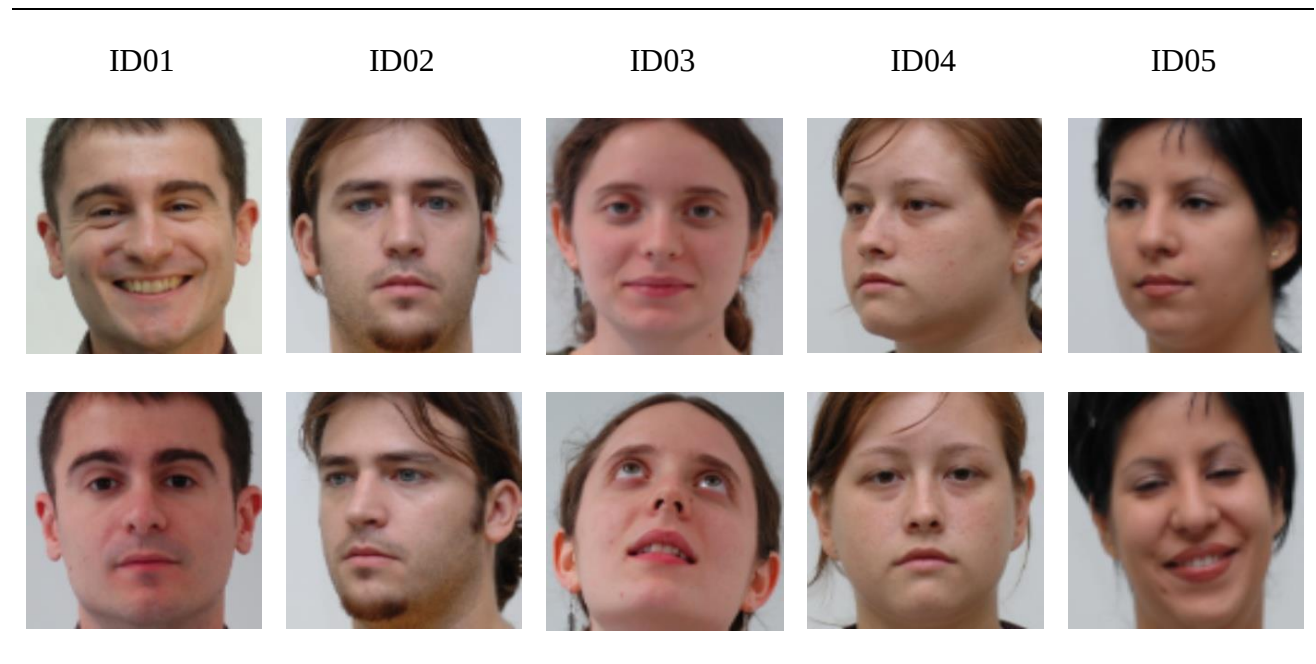


Figure 5. 1 Sample Detected Faces on GTAV database

The following Figure shows sample images from Self-prepared face database after face detection.

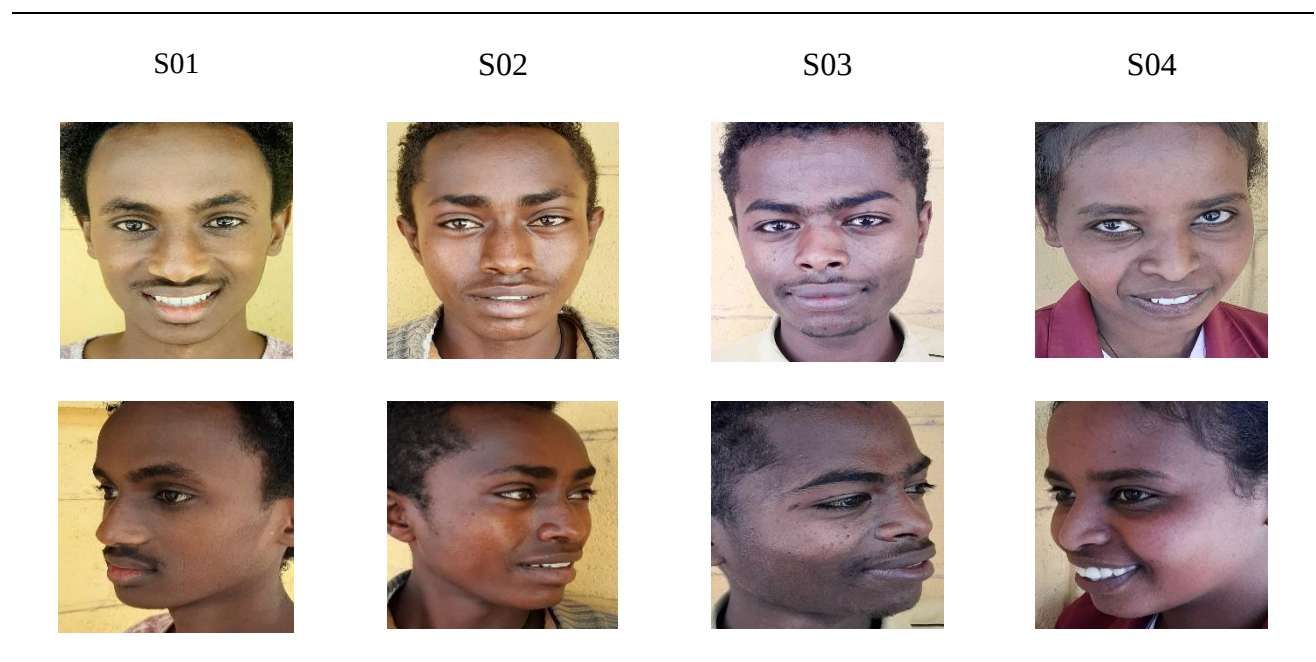


Figure 5. 2 Sample Detected Faces on Self-prepared face database

One of the augmentation technique used is Horizontal Flip Augmentation, where the columns of pixels of detected face image is reversed horizontally. The following figure shows code snippet for flipping the face image.

```
1
2 def flip_image(theImage):
3     # Flip image from left to write
4     flippedImage = np.fliplr(theImage)
5
6     # return the flipped image
7     return flippedImage
```

Figure 5. 3 Code Sample for Horizontal Flip Augmentation.

The numpy method called fliplr used to flip the image. It flips an array in the left or right direction. Flips the entries in each row in the left or right direction where the columns are preserved, but appear in a different order than before. The following figure shows sample converted images.



Figure 5. 4 Sample Images of GTAV database after Horizontal Flip Augmentation

The following figure shows Horizontal Flip Augmentation applied on Self-prepared face database.

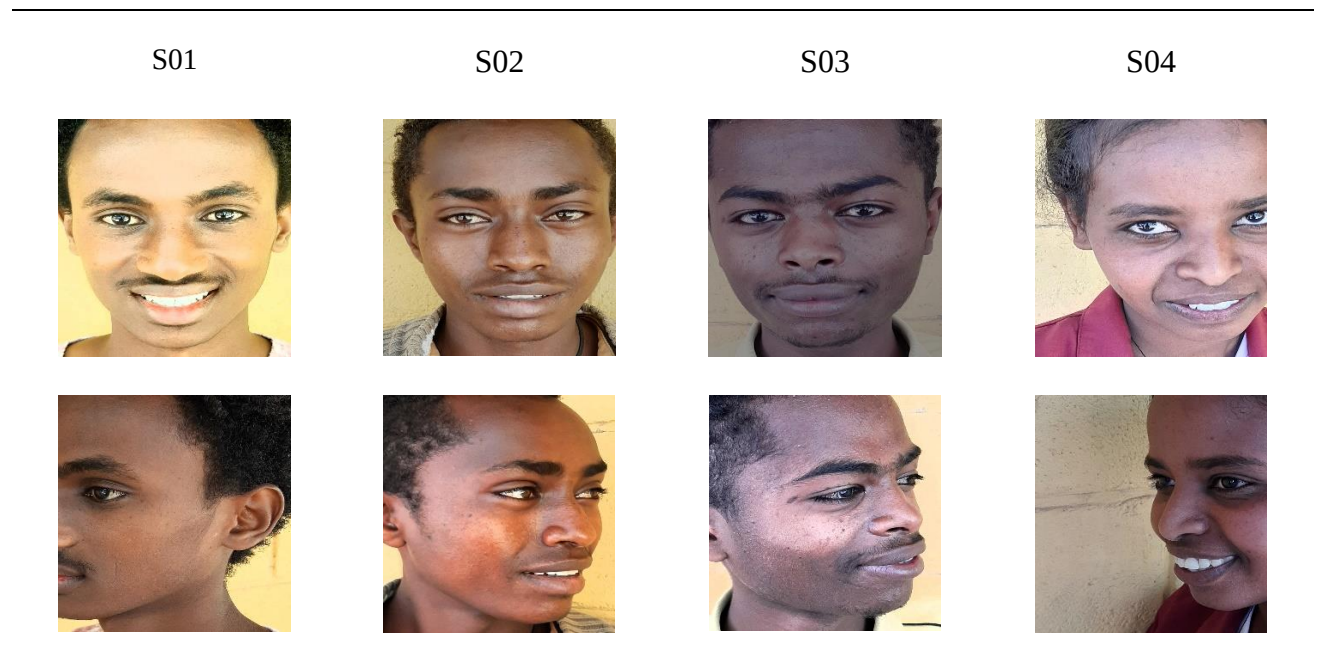


Figure 5. 5 Sample Images of Self-prepared database after Horizontal Flip Augmentation

The other augmentation mechanism applied is adding random brightness. Random Brightness added to the detected face by converting the image to HSV Color model. Figure 5.4 code snippet show the method used to manipulate the images brightness.

```
1 def brightness_augment(img, factor=0.5):
2     #convert to hsv
3     hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
4     hsv = np.array(hsv, dtype=np.float64)
5     #scale channel V uniformly
6     hsv[:, :, 2] = hsv[:, :, 2] * (factor + np.random.uniform())
7     #reset out of range values
8     hsv[:, :, 2][hsv[:, :, 2] > 255] = 255
9     rgb = cv2.cvtColor(np.array(hsv, dtype=np.uint8), cv2.COLOR_HSV2BGR)
10    return rgb
```

Figure 5. 6 Code Sample for Brightness Augmentation

The following figure shows sample of the images in GTAV database after brightness augmentation applied.



Figure 5. 7 Sample Images of GTAV database after Random Brightness Augmentation

The following Figure shows sample of the Self-prepared database images after random brightness augmentation applied.

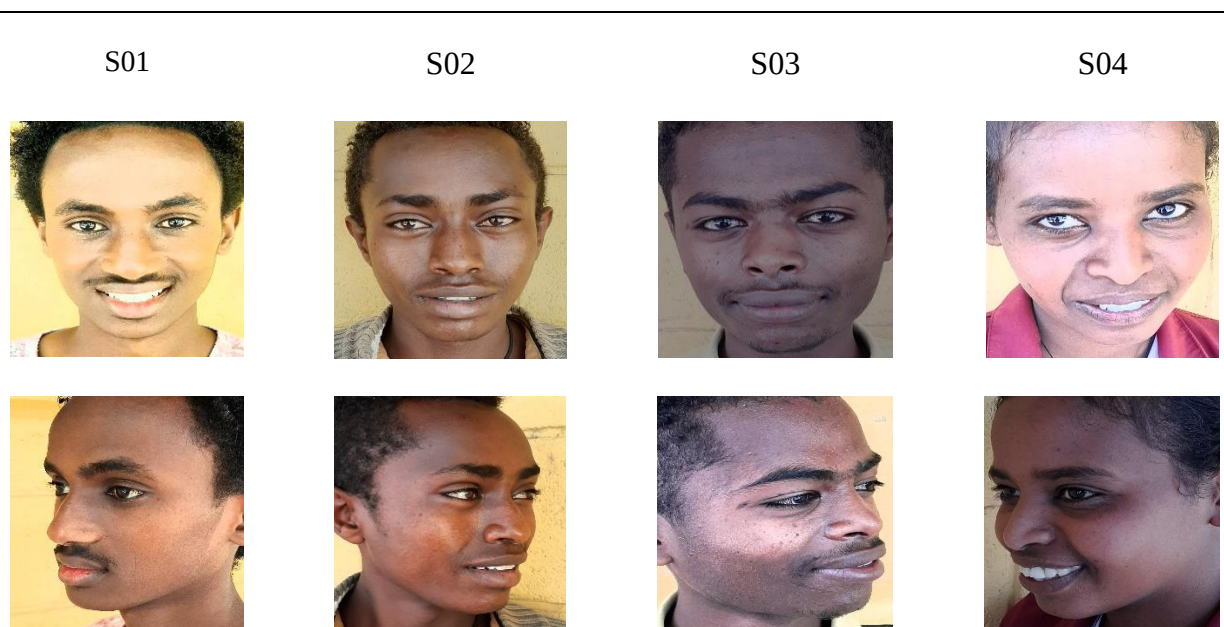


Figure 5. 8 Sample Images of Self-prepared database after Random Brightness Augmentation

Another method used for augmentation is Gaussian Blur, where OpenCV method called Gaussian Blur is used. For applying Gaussian blur 11 by 11 filter used as shown in figure 5.8.

```

1
2 def blur_image(theImage):
3     # apply gaussian blur
4     blurredImage= cv2.GaussianBlur(theImage, (11,11), 0)
5
6     # return the blurred image
7     return blurredImage

```

Figure 5. 9 Code Snippet for Gaussian Blur Augmentation.

The Gaussian blur smooth the image like shown in the bellow figure, where it has sample images of the dataset after applying the Gaussian blur.



Figure 5. 10 Sample Images after Gaussian Blur Augmentation on GTAV database

From GTAV Face Database about 1204 images of 44 classes used by this study, which is about 27 images per class on average. After augmentation, the number of images is grown by 300 percent, which is total of 3612 images and 82 images per class on average. The following Figure shows images after Gaussian Blur Augmentation for Self-prepared face database.

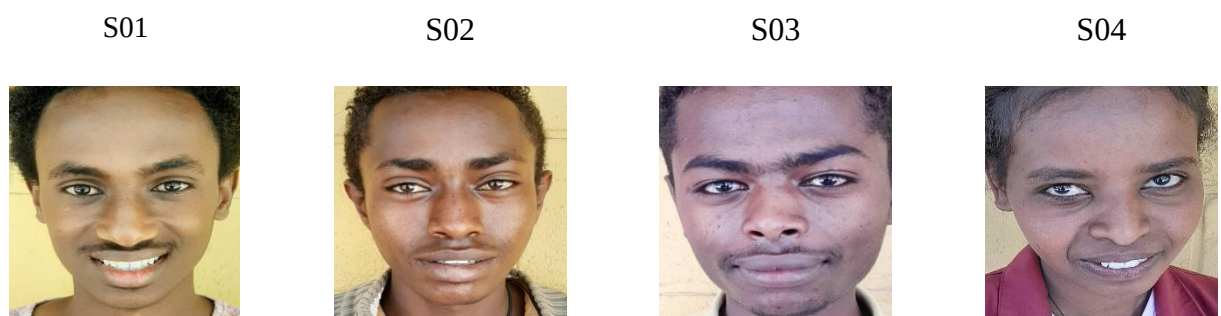




Figure 5. 11 Sample Images after Gaussian Blur Augmentation on Self-prepared database

From Self-prepared Face Database about 80 images of 20 classes used by this study, which is four images per class. After augmentation, the number of images grown by 300 percent, which is total of 297 images and about 15 images per class.

5.4 Face detection and Facial Component Extraction Implementation

Face recognition algorithm takes two inputs, the detected face and the facial component (like nose, mouse, eyes etc.), so the face extraction have two steps where first the face is detected and then the facial component is extracted from the detected face. Both the face detection and facial component extraction is performed using pre-trained MTCNN model.

This section presents the implementation of the face detection and facial component extraction methods shows in figure below.

```
4 # MTCNN face detectin module intialization
5 detector = MTCNN()
6
7 # face detection using MTCNN
8 faces = detector.detect_faces(image)
```

Figure 5. 12 Sample Code for Initializing the Face Detection Module and Face Detection

The face detector takes an image (numpy array) as an input and detects the face and facial feature, and outputs an array containing the location of the face (or faces if there is more than one face in the image) and the location of the facial features.

```
{'box': [1942, 716, 334, 415], 'confidence': 0.9999997615814209,
'keypoints': {'left_eye': (2053, 901), 'right_eye': (2205, 897),
'nose': (2139, 976), 'mouth_left': (2058, 1029), 'mouth_right':
(2206, 1023)}}
```

Figure 5. 13 Sample Face and Facial Feature Detection Output

5.4.1 Face Detection

The detected face extracted from the output of the face detection model by iterating over the detection output. The detected face can be ignored based on the model confidence that the detected object is face or not. The figure 5.10 below show code for detected face extraction.

```
10 # extract detected faces
11 for result in faces:
12     # get coordinates
13     x, y, w, h = result['box']
14     cv2.rectangle(image, (x-10, y-15), (x+w+15, y+h+15), (0, 255, 0), 2)
```

Figure 5. 14 Code Snippet for Detected Face Extraction

The output of the face detector is a map or dictionary so that face and facial component extracted by calling the key. Each face in the image have two parts, box and key-points. The box contains the position of the face in the image and the key-points are contains nose, mouse and eyes.

The face extracted by adding 10 and 15 pixels in x-axis and y-axis of the image, in order to add more part of the face, since the detected face is tight. The detected face cropped from the image and saved for face recognition.

5.4.2 Facial Component Extraction

The facial component extracted from the single face in the image are the nose, right and left point of the mouse, and both the eyes. The facial components represented in the output of the face detection model by key-points keyword, where it is a dictionary by itself.

```
12 p = 0
13 for key, (x,y) in result['keypoints'].items():
14     theImage = cv2.resize(image[y-10:y+10, x-10:x+10], (56,56))
15
16     if(p == 0):
17         im_h = theImage
18         p = + 1
19     else:
20         im_h = cv2.addWeighted(im_h,0.5,theImage,0.5, 0)
```

Figure 5. 15 Code Snippet for Facial Component Extraction

The dictionary includes the centre position of each component in the face, so rectangle constructed from the centre position by adding and subtracting 10 pixels from each direction.

The extracted facial components added together using weighted Sum of 0.5 and resized to 56 by 56 image to create one feature then image saved for face recognition step and will be fed to the face recognition model.

5.5 Face Recognition Implementation

The face recognition implementation includes two steps, feature extraction, and classification steps. Two deep learning frameworks used to implement the deep CNN model for face recognition Tensorflow and Keras (with Tensorflow backend).

Google Colab used to train the model since the training deep learning architecture takes a lot of time. Google Colab provides a 135 GB Disk and 12 GB RAM with GPU free. The developed model tested on google Colab. In order to train the model, all the dataset uploaded to the google drive and accessed or mounted on the google Colaboratory.

5.5.1 Facial Feature Extraction Implementation

For the feature extraction from the face image MobileNetV2 is used, which is available in one the popular deep learning library Keras. The feature extractor takes two inputs, detected face and facial component. The detected face image extracted in the pre-processing stage fed into the pre-trained mobile net architecture. The MobileNetV2 can be loaded using Keras library as shown in the following figure code snippet.

```
1 mobilev2 = keras.applications.mobilenet_v2.MobileNetV2()  
2 mobilev2.summary()
```

Figure 5. 16 Code Snippet for Loading MobileNetV2

After loading the MobileNetV2, the last three layers removed, since it contracted and trained for 1000 classes and our class is 44. The trainability of all but the last 26 Layers are set to false to fine tune only the last 26 layers of the architecture to the prepared dataset. The figure bellow show code for set trainable parameters as below.

```
1  
2 x = mobilev2.layers[-3].output  
3 mobile = Model(inputs=[mobilev2.input], outputs=x)  
4  
5 for layer in mobile.layers[:-26]:  
6     layer.trainable = False  
7
```

Figure 5. 17 Code Snippet for Set Trainable Parameters.

The second input is the facial component from the detected face. The following figure 5.18 code shows the architecture construction for feature extraction of the facial component. The architecture contains four 2D convolutional Layers and Batch normalization. It takes an input of 56 by 56 image and after performing three max pooling, the output of the feature extractor is 7 by 7 by 64 feature.

```

1
2 input1 = keras.layers.Input(shape=(56, 56, 3))
3 land = Conv2D(32, (3,3), padding='same', activation="relu")(input1)
4 land = BatchNormalization()(land)
5 land = MaxPooling2D()(land)
6 land = Conv2D(64, (1,1), padding='same', activation="relu")(land)
7 land = BatchNormalization()(land)
8 land = MaxPooling2D()(land)
9 land = Conv2D(64, (1,1), padding='same', activation="relu")(land)
10 land = BatchNormalization()(land)
11 land = MaxPooling2D()(land)
12 land = Conv2D(64, (1,1), padding='same', activation="relu")(land)
13 land = BatchNormalization()(land)

```

Figure 5. 18 Code Snippet for Facial Component Feature Extractor

The feature extractor of the detected face and facial component concatenated to construct the feature extractor. In order to train the feature extractor a softmax classification layer with 44 classes added.

```

16 x = mobile.layers[-1].output
17 x = Conv2D(128, (1,1), padding='same')(x)
18
19 x = keras.layers.Concatenate()([x, land])
20 x = Conv2D(32, (1,1), padding='same', activation="relu")(x)
21 x = BatchNormalization()(x)
22 x = GlobalAveragePooling2D()(x)
23 predictions = Dense(44, activation="softmax")(x)
24 model = Model(inputs=[mobile.input, input1], outputs=predictions)

```

Figure 5. 19 Code Snippet for Model Construction Feature Extraction

5.5.2 One-Shot Implementation

This research paper uses One-shot learning techniques by using the pre-trained feature extractor trained on prepared dataset. The one shot learning is implemented using Siamese Neural Network, which consist of two identical networks sharing the same parameters since both images should be extracted in the same way.

```

6 input_shape = (224, 224, 3)
7 input_shape_land = (56, 56, 3)
8
9 input1 = Input(input_shape)
10 input2 = Input(input_shape_land)
11
12 left_input = Input(input_shape)
13 right_input = Input(input_shape)
14
15 left_input_land = Input(input_shape_land)
16 right_input_land = Input(input_shape_land)

```

Figure 5. 20 Code Snippet for Siamese Network Input Layers Initialization

Since the feature extractor takes two inputs, two inputs initialized and then four input layer are initialized two for each identical network.

```

28 convnet = mobile([input1, input2])
29
30 theModel = Model(inputs=[input1, input2], outputs=convnet)
31
32 encoded_l = theModel([left_input, left_input_land])
33 encoded_r = theModel([right_input, right_input_land])
34
35 merge_layer = Lambda(euclidean_distance)([encoded_l, encoded_r])
36 merge_layer = Flatten()(merge_layer)
37 merge_layer = Dense(100)(merge_layer)
38 merge_layer = Dense(100)(merge_layer)
39 dense_layer = Dense(1, activation="sigmoid")(merge_layer)
40 model = Model(inputs=[left_input, left_input_land, right_input,
41                      right_input_land], outputs=dense_layer)
42

```

Figure 5. 21 Code Snippet for One Shot Model Construction

The two identical networks created by using pre-trained feature extractor and padding the four input layers, two for each network. The figure 5.18 below show the implementation of the architecture of One Shot using the Siamese Network

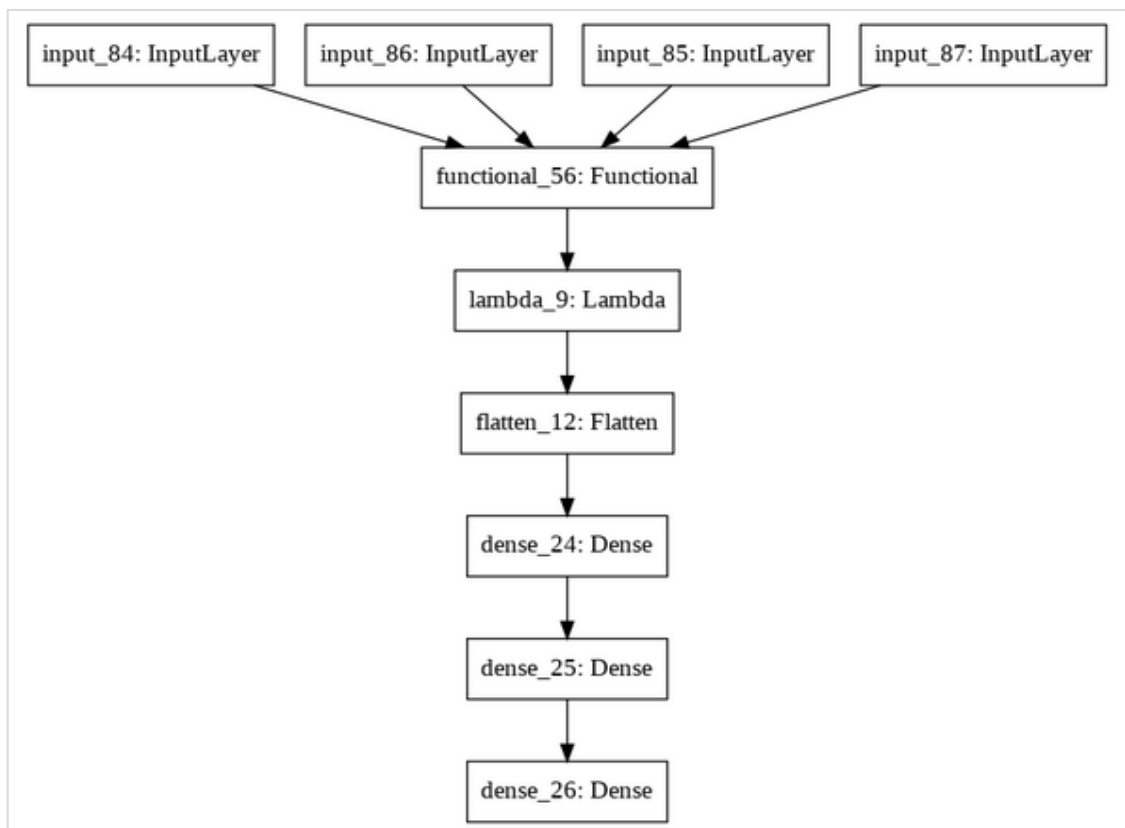


Figure 5. 22 The Implementation of One Shot Architecture using Siamese Network

The output of the identical networks passed to the Euclidean distance loss function, which calculates the similarity of the features passed by the feature extractor. The following figure code snippet shows the implementation of the Euclidean distance.

```
1 def euclidean_distance(vects):
2     x, y = vects
3     sum_square = K.sum(K.square(x - y), axis=1, keepdims=True)
4     return K.sqrt(K.maximum(sum_square, K.epsilon()))
```

Figure 5. 23 Code Snippet for Euclidean Distance Implementation as Loss Function

The output of the above operation is passed to two dense layer and then to the binary classification layer. The Functional name represents the pre-trained feature extractor.

5.6 Evaluation Implementation

In order to measure the performance of the model a confusion matrix is computed. The confusion matrix is calculated from the predicted label and the truth-value. Figure 5.24 code snippet show the matrix creation.

```
1 cm = confusion_matrix(np.argmax(valid_label_data, axis=1), prediction)
```

Figure 5. 24 Code Snippet for Matrix Creation

Figure 5.25 code snippet shows the implementation of evaluation metrics. The evaluation metrics include the Recall, Precision, Accuracy and F1 Score.

```
1 # Recall = TP / (TP + FN)
2 yishak_recall = np.diag(confMat) / np.sum(confMat, axis=1)
3
4 # Precision = TP / (TP + FP)
5 yishak_precision = np.diag(confMat) / np.sum(confMat, axis=0)
6 h = 0
7 for i in range(44):
8     h = h + confMat[i][i]
9 acc = h/len(all_labels)
10
11 yishak_f1 = (2 * yishak_recall * yishak_precision) / (yishak_recall + yishak_precision)
12
13 print("Accu = " + str(np.mean(acc)))
14 print("Recall = " + str(np.mean(yishak_recall)))
15 print("Precision = " + str(np.mean(yishak_precision)))
16 print("F1 = " + str(np.mean(yishak_f1)))
```

Figure 5. 25 Code Snippet for Recall, Precision, Accuracy and F1 Score

Chapter Six

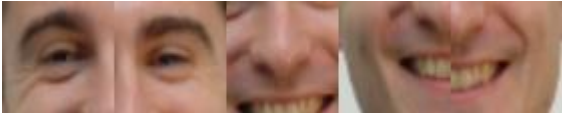
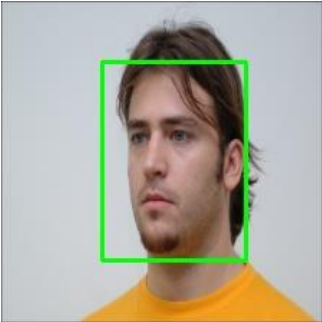
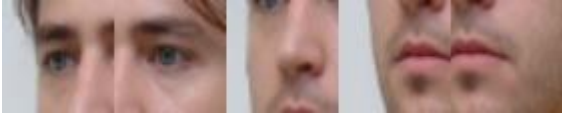
6 Evaluation, Results and Discussion of the Study

6.1 Overview

This chapter presents the evaluation of the proposed solution and overall result of the study. It also presents the ablation study performed to see the effects of the components of the model. The chapter also discusses the result of the recognition model by comparing it to related works to evaluate the approaches used in the proposed model and to check whether the method is robust or not.

6.2 Face Detection

Table 6.1 shows the Sample face Detection and Facial Component result Images for first four classes in the GTAV database. The left image show the detected face using MTCNN and the right images show the facial components extracted from the detected face image.

Sample Face Detection and Facial Component Images for GTAV database		
Classes	Face Image	Facial Components
ID01		
ID02		

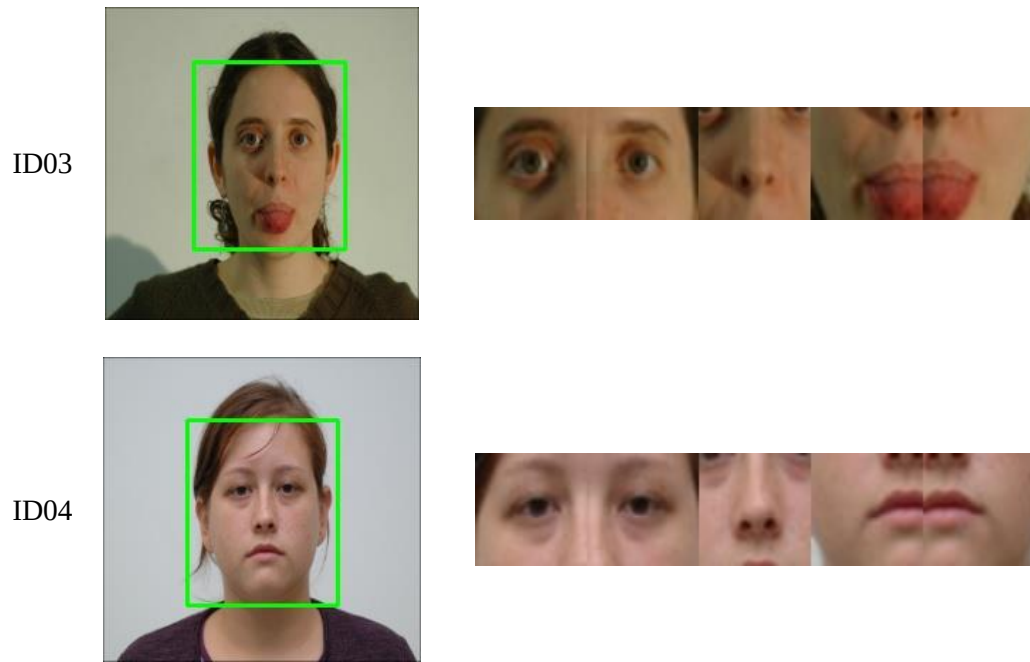


Table 6. 1 Sample Face Detection and Facial Components Images in GTAV database

Table 6.2 below shows the Sample face Detection and Facial Component result Images for four classes in Self-prepared database. The left image show the detected face using MTCNN and the right images show the facial components extracted from the detected face image.

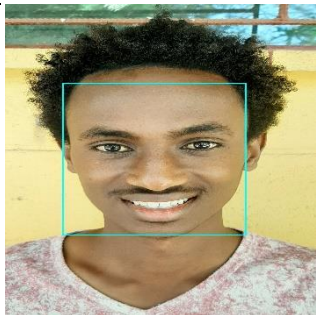
Sample Face Detection and Facial Component Images for Self-prepared database		
Classes	Face Image	Facial Components
S01		



Table 6. 2 Sample Face Detection and Facial Components Images in Self-prepared database

6.3 Face Recognition

6.3.1 Feature Extractor

The purpose of the feature extractor is to represent the relevant features of each image accurately for Siamese architecture. The feature extractor takes the face image and facial component of single person in the database or query image and extract the relevant feature for calculating the similarity score. In order to train the feature extractor a softmax classification layer added. The image dataset contains 3612 images from GTAV database (Augmentation) and the image dataset contains 297 images from Self-prepared database (Augmentation).

For training 92% of the total dataset images are used and 5% for validation during training and for testing about 3% of the dataset is used since cross-validation technique is Holdout. The batch size is 32 and 50 epoch chosen to train the network, since the number of images is small and the model converges with chosen parameter. The following figure 6.1 show the train and test accuracy and loss of the model during training step.

The model trained with learning rate of 0.0001 with Adam optimizer and Categorical Cross entropy loss. Different learning rate is applied to choose the best fitting learning rate, but when lower learning rate is used, the model does not converge since it needs more time and data to train and when higher learning rate is used the model learns fast but the accuracy is not stable.

The trained model achieves better accuracy on both training and validation dataset as shown in the bellow figure 6.1 a. On training dataset the model reaches 100% accuracy on the 16th epoch and 23rd epoch on the testing dataset even though it tend to get lower accuracy on some epochs as shown in the figure 6.1.

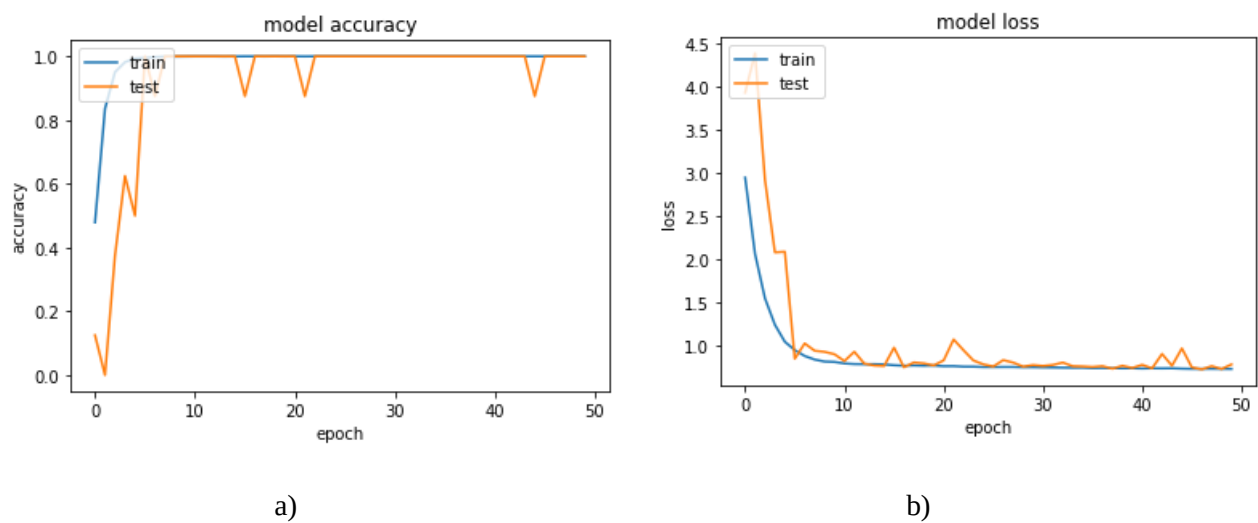


Figure 6. 1 Feature Extractor Accuracy and Loss during Training

The same reflected in the figure 6.1 b, when the accuracy decreases the loss is higher. The model converges to 100% accuracy on both training and validation set. The confusion matrix in the figure 6.2 below shows the matrix of the first 10 classes (the full confusion matrix in the appendix) on the test set.

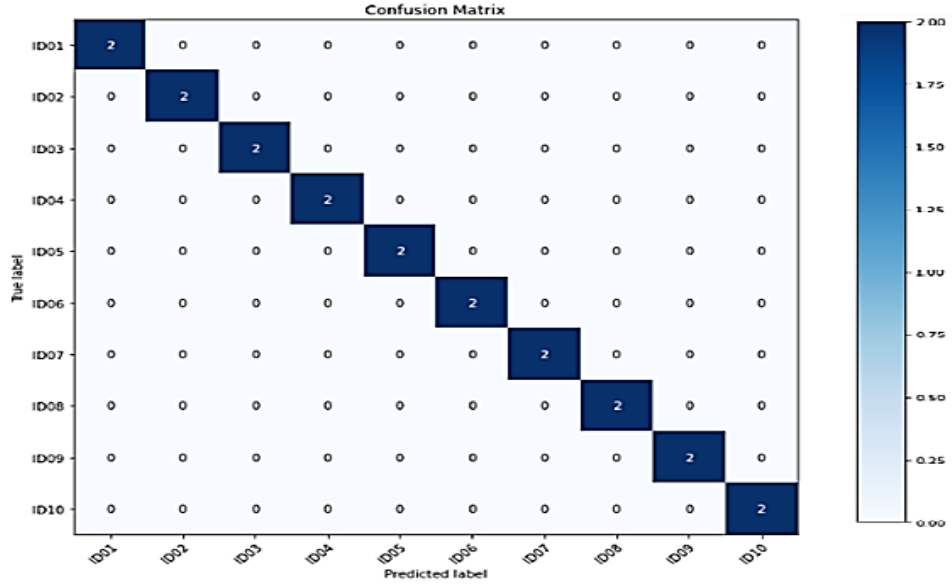


Figure 6. 2 Confusion Matrix of the Feature Extractor on Test Set

As shown in the above figure 6.2 all images in the test set is correctly classified. The accuracy, recall, precision, and F1 score of the model is 100%. The same model can be used for face recognition on the trained 44 classes. When another class (45th class) or person is needed to be added the model should be trained again, since it only trained on 44 classes. To solve this one shot learning technique is applied using Siamese network where only one image of the person in the database to recognize.

6.3.2 One-shot Model Result

The One-shot model takes two images, query image and database image, and extracts the feature of both images using two identical networks. Then the similarity score is calculated from the extracted feature. The feature extractors manipulate the image the same way since they share the same parameter. Since the one shot model is composed of the feature extractor and similarity function, the pre-trained feature extractor with accuracy of 100%.

For the training of the model, 92% of the dataset is used as training set, and 5% of the dataset as validation set, and 3% for test set. Since the model outputs the probability that the two images are similar; it is a binary classification problem. For training the model, the dataset is organized into pair images where each image in the dataset is paired with positive (similar) and negative (no similar) image. Therefore, the number of images used for training and validation is doubled. 92% of the dataset is about 3250 images and after pairing the images it is convert into training set of 6500 images. The validation set contains 181 images (5% of the dataset) and after pairing it is converted into validation set of 362 images. The label of each paired image is set as one or zero, if the paired images are of the same class or not respectively.

Figure 6.3 shows the One-shot model Training accuracy and loss on training and testing set. The learning rate of Adam Optimizer used is 0.0001 with Binary Cross Entropy (since the labels are binary) and the number of epoch is 500. In order to avoid overfitting checkpoint is used. Only the weight with best accuracy and loss is saved.

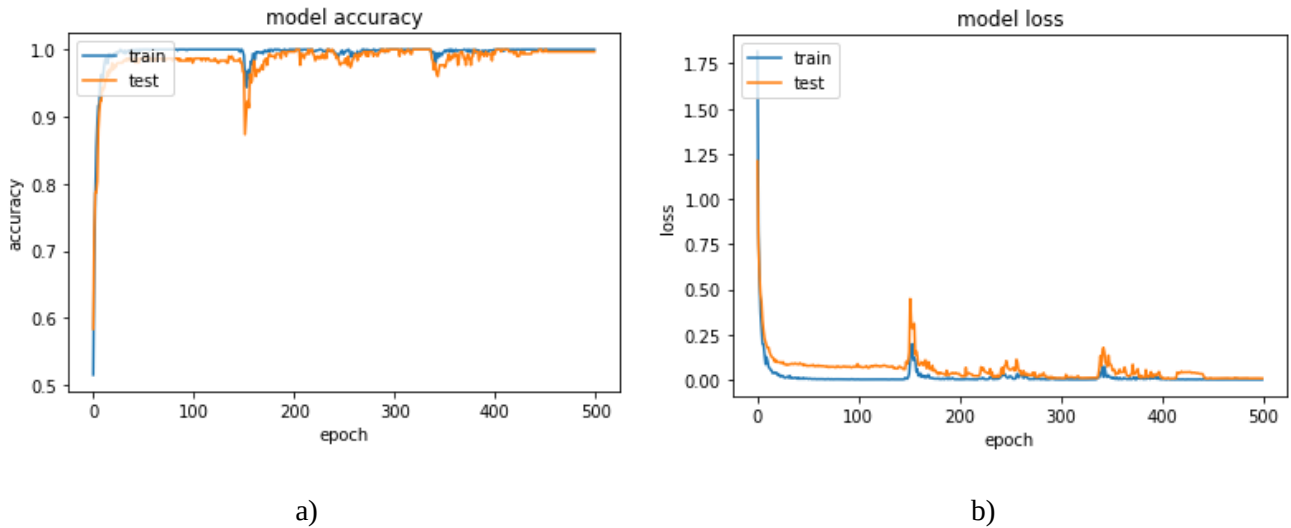


Figure 6. 3 One-shot Accuracy and Loss on Training and Testing Set

The model converges to 100% accuracy starting from 450 epoch (there are some epochs where the accuracy is 100%, but it is not stable). Figure 6.4 shows the confusion matrix of One-shot model for test set of the first 10 classes in GTAV database (full confusion matrix for 44 classes can be found in the Appendix). The test set contains 88 images (about 3% of the dataset) two images for each class, where each image compared with database image (one image for each class).

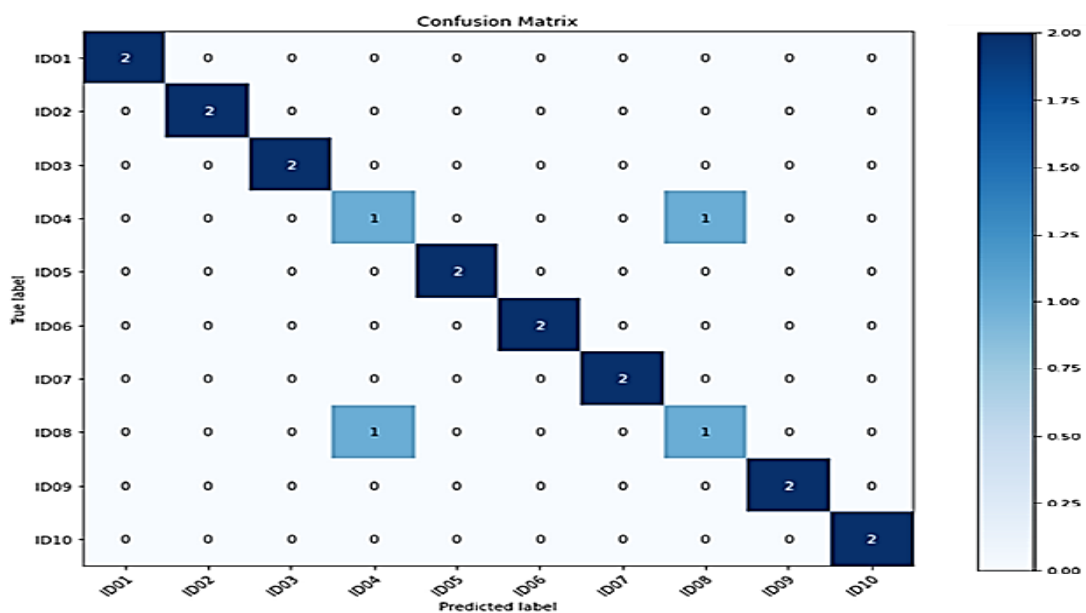


Figure 6. 4 Confusion Matrix for One-shot Model in GTAV database

As shown in the above figure 6.4, the model misclassify the four images for class ID04, ID08, ID31 and ID33 (the last two are in the Appendix). The model achieves 95% accuracy, precision, recall and F1 Score. In the Test stage, the model generates the similarity score of the query image with each image in the database, and then the query image is classified as the one with highest similarity score. If the highest score is not with the correct class, the second and third highest score can be the correct class. One of the advantage of the proposed solution is that it can be used for retrieving images from database with highest score.

Figure 6.5 below shows the confusion matrix of One-shot model for test set of the first 14 classes in Self-prepared database (full confusion matrix for 20 classes can be found in the Appendix). The test set contains about 20% of the dataset two images for each class, where each image compared with database image (one image for each class).

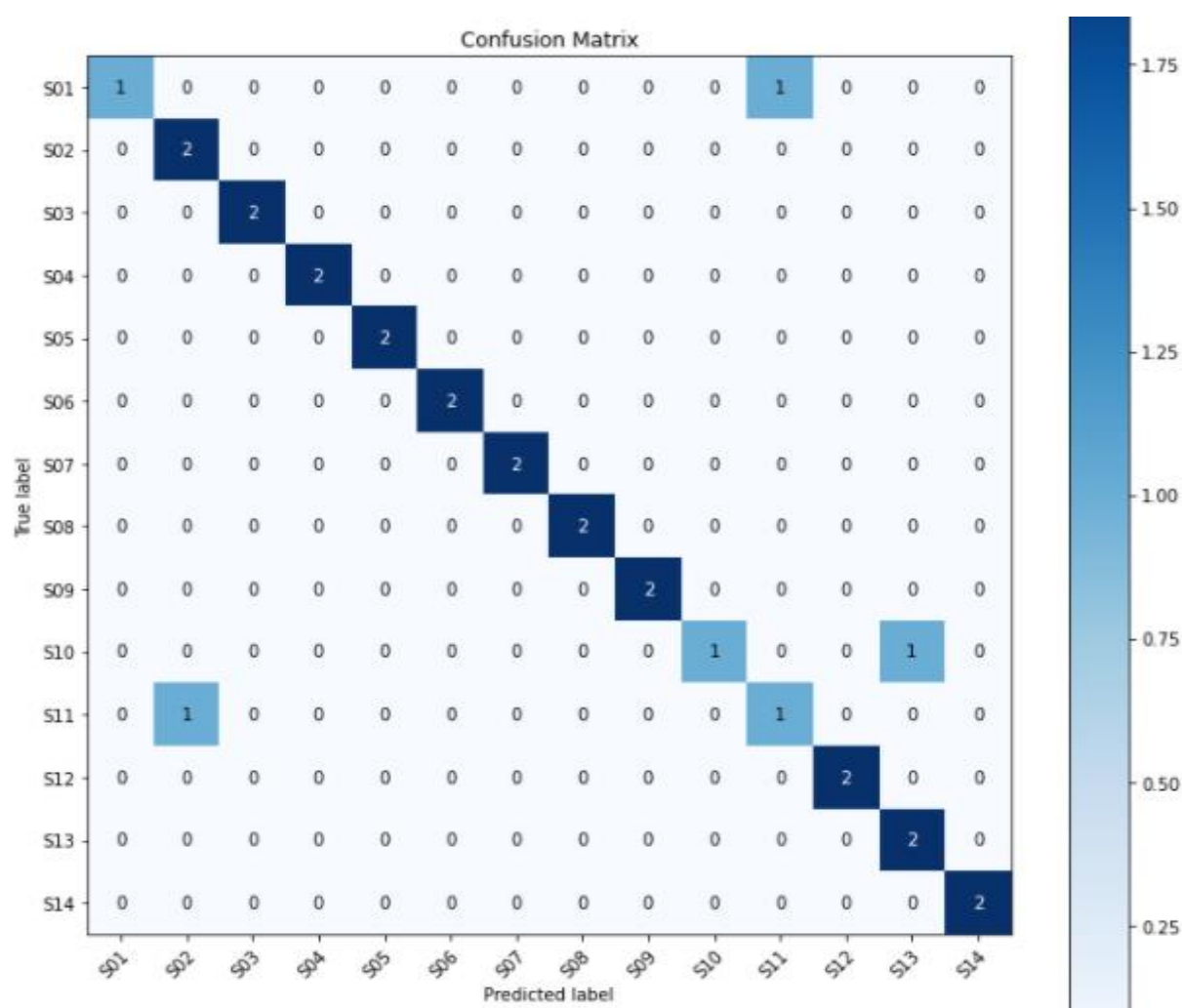


Figure 6. 5 Confusion Matrix for One-shot Model in Self-prepared database

The above figure shows that the model misclassify three images for class S01, S10, and S11 (the full confusion matrix is in the appendix). The model achieves 93% Accuracy, 94% Precision, 93% Recall

and 92% F1 Score. In the Test stage, the model generates the similarity score of the query image with each image in the database, and then the query image classified as the one with highest similarity score. If the highest score is not with the correct class, the second and third highest score can be the correct class.

Different One-shot (OS) model created for checking the effect of different components of the model. The table below shows different One-shot models created to experiment purpose such as One-shot model without Autoencoder (OS-AE), One-shot model with Autoencoder (OS+AE) and One-shot model without Components (OS-Comp).

Model	Input Image	Training	Validation	Test	Database
<i>OS-AE</i>	3612	92%	5%	3%	GTAV
<i>OS+AE</i>	3612	92%	5%	3%	GTAV
<i>OS-Comp</i>	3612	92%	5%	3%	GTAV

Table 6. 3 One-shot models used for Experiment

All the data divided into three subsets in this study; training set, validation set and test set. Then apply different splitting technique 70/30, 80/20, 90/10 but overfitting is occur because of the dataset size is small. Finally, the researcher split 92% for training, 5% for validation, and 3% for test.

The first one is one shot model without pre-trained feature extractor (auto-encoder, AE), which is basic Siamese network. The model trained with the same dataset and 300 epochs and a validation accuracy of 97% is achieved. Even though the validation accuracy in the training stage is good, the testing accuracy is not as well. The testing accuracy achieved on this network is 0.36, which is not even close enough for real world application. Figure 6.5 shows the training and validation loss and accuracy of the first model.

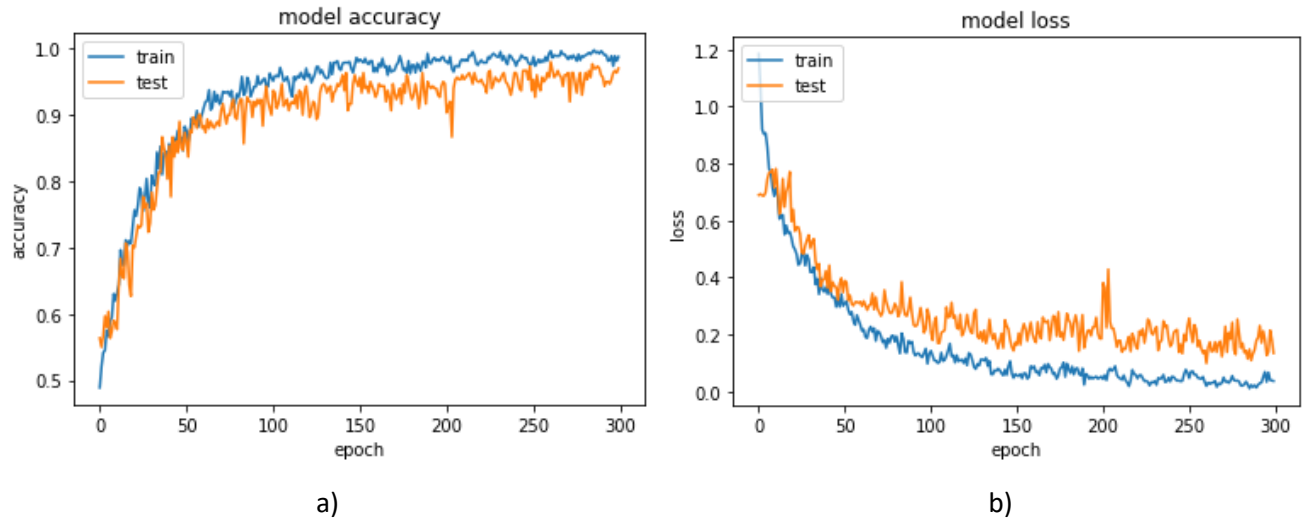


Figure 6. 6 One-shot Training Accuracy and Loss of OS - AE

The second model is Siamese network with auto-encoder (a pre-trained feature extractor that created from scratch). The feature extractor is trained on the whole dataset to teach how to extract relevant feature from the images. The pre-trained feature extractor is used as two identical networks in the Siamese Network (one-shot model). The figure 6.6 shows the training and validation loss and accuracy of the second model.

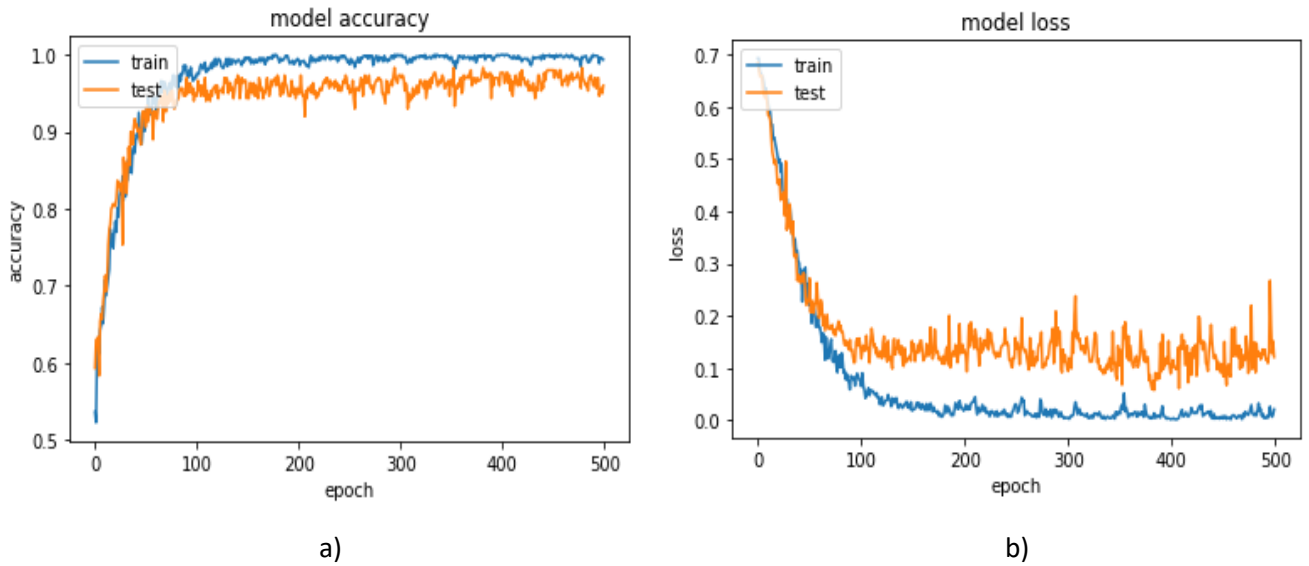


Figure 6. 7 One-shot Training Accuracy and Loss of OS + AE

In this configuration, the model trained with 500 epoch on the same dataset and train-test split. An accuracy of 0.875 or 87% achieved on the test set.

The third model is One-shot model takes two images, query image and database image, and extracts the feature of both images using two identical networks. The similarity score is then calculated from the extracted feature. The same as the proposed model but in this model the feature extractor takes the

face image only which means without facial components or facial landmarks. The testing accuracy achieved on this network is 0.58. The figure 6.7 shows the training and validation loss and accuracy of the third model.

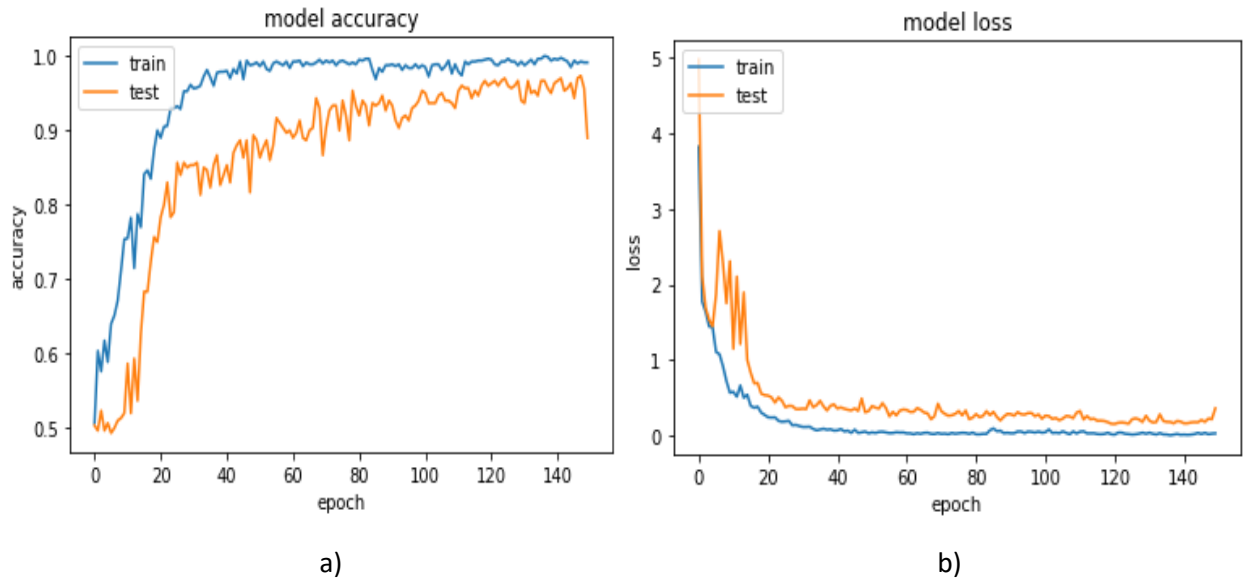


Figure 6. 8 One-shot Training Accuracy and Loss of OS – Comp

The following Table shows the comparison of different One-shot models that used in this research.

<i>Comparison of Different One-shot Models</i>				
<i>Models</i>	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1</i>
OS - AE	0.34	0.34	0.36	0.33
OS + AE	0.875	0.875	0.902	0.870
OS - Comp	0.579	0.579	0.579	0.579
OS + MN	0.954	0.954	0.954	0.954

Table 6. 4 Comparison of Different One-shot Models

The fourth model is the proposed model, which is One-shot model with the combination of MobileNetV2 and Siamese Network (OS+MN). The above Table shows the results obtained from the four models on the same test set. The bold line shows the best performing model on the test set. The proposed model out performs the other three as shown on the above Table 6.4.

Papers	Result (Accuracy)
Face Recognition Based on Deep Learning	91% on ORL and 81% on LFW database
Component-Based Face Recognition Under Transfer Learning for Forensic Applications	98%
Image-based Face Detection and Recognition	88%
Wavelet Transform based Feature Extraction for Face Recognition	80%
Facial Soft Biometric Features for Forensic Face Recognition	100% for ATVS & 75% MORPH database
(OS+MN) Proposed model	100% on classification model and 95% for One-shot model for GTAV & 93% for Self-prepared database

Table 6. 5 Related Work comparison with Proposed Model

As shown in the above table 6.5 the second and fifth method out performs the current model in a recognition rate, but such models are not efficient in the real world since its need a training if additional class added and needs many data to train. The proposed model is One-shot model, so it only needs one image in the database to recognize a face of a person. Whenever a class is added, an image representing the face of that person is added in the database and using the One-shot model the query image can be used to search through database.

The following table shows sample result images of the first four classes in GTAV database. The images show the detected face and the top three classes with the similarity score. In the confusion matrix of the One-shot model only the top one results are used, so if the top two or three result were used the model will have less misclassification.

Classes	Sample Face Recognition Images		
ID01			
ID02			
ID03			
ID04			

Table 6. 6 Sample Images of Face Recognition in GTAV database

The following table shows sample result images of four classes in Self-prepared database. The images show the detected face and the top four classes with the similarity score or less misclassification.

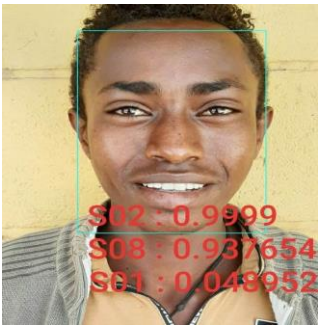
Classes	Sample Face Recognition Images		
S01	 S01 : 1.0 S07 : 0.986799 S08 : 0.968989	 S01 : 0.999937 S07 : 0.79745 S13 : 0.689395	 S01 : 0.999967 S03 : 0.997654 S11 : 0.689895
S02	 S02 : 0.9991 S01 : 0.98765 S06 : 0.6898952	 S02 : 0.9999 S08 : 0.937654 S01 : 0.048952	 S02 : 1.0 S07 : 0.947654 S01 : 0.786895
S03	 S03 : 0.9999677 S20 : 0.0086545 S16 : 1.3498952	 S03 : 0.9898677 S19 : 0.8936545 S12 : 0.8898952	 S03 : 0.9993277 S13 : 0.4576545 S11 : 0.258952
S04	 S04 : 1.0 S08 : 0.037654 S18 : 0.034895	 S04 : 1.0 S13 : 0.699654 S09 : 0.588952	 S04 : 0.9998 S03 : 0.6965 S11 : 0.0198

Table 6. 7 Sample Images of Face Recognition in Self-prepared database

6.4 Output of the Study

In order to achieve the research objective, an experimental testbed designed and a model for face recognition was developed. The model focused on finding a solution to the research problem in the section 1.4.

To answer research question 1, in the literature review and related work chapter, different Face Recognition techniques identified. Finally, MobileNetV2 and Siamese Neural Network methods are selected for face recognition. The MobileNetV2 architecture uses separable convolutions to reduce the number of parameters because of the reason to choose. Compared to most architectures, it has less parameters and operations, which makes it able to run fast on various devices with less computing capabilities. Due to the reason to choose Siamese Neural Network is to find the similarity of the inputs by comparing their feature and it is no need to retrain when new data added to the dataset.

To answer research question 2, the researcher make different experiments by using Facial Landmarks and without using Facial Landmarks or Components, and then by compare them. Therefore, the result shows that using Facial Landmarks has get better accuracy than face recognition without Facial Landmarks, the comparison shown as in Table 6.4.

To answer research question 3, the researcher used GTAV face database and Self-prepared face database that contains different pose variation, different facial expression and different illumination. Then the experiments implement on these two databases, and the result shows in section 6.3.2 is that Facial Landmark-based face recognition can solve pose variation, expression, and illumination problems. This proposed research is useful for forensic face recognition, because of our model is more accurate than the others are related works that worked in Forensic Face Recognition.

Chapter Seven

7 Conclusion and Future Work

This chapter presents the conclusions drawn from the results of the experiment and the researcher's advice on further actions that should be taking, and future research areas.

7.1 Conclusion

The main objective of this study is to develop a Hybrid Component-based Face Recognition technique to build a reliable and fast face recognition algorithm to enhance the performance of Forensic Face Recognition.

The proposed face recognition algorithm is a combination of different steps. The first one is face and facial landmarks detection, which is responsible for providing the detected face and facial landmarks. For the detection of face and facial landmarks, the Multi-task Cascaded Convolutional Neural Network (MTCNN) is used. The second step is pre-processing of image; the researcher used image resize and augmentation. Different types of augmentations used such as Horizontal Flip augmentation, Random Brightness augmentation, and Gaussian Bluer augmentation. The third step is Feature Extraction. For feature extractor MobileNet-v2 Convolutional Neural Network applied. The model trained with 3,612 images from the GTAV Face Database on training dataset the model reaches 100% accuracy. Lastly, Siamese Neural Network used to learn whether two faces are similar or dissimilar. This proposed research for face detection and recognition is try to improve or increase the degree of correctness.

Experiments were conducted on the GTAV face database and Self-prepared face database with different positions, illuminations, and facial expressions. In GTAV face database, the model achieves 95% Accuracy, Precision, Recall and F1 Score. The experimental results indicate that the developed face recognition algorithm has better performance than the related works. The model obtained 93% Accuracy, 94% Precision, 93% Recall and 92% F1 Score in the Self-prepared face database.

The experimental results are compared with those of One-shot with Autoencoder, One-shot without Autoencoder, and One-shot without Components in the same database, the researcher can see that One-shot with MobileNetV2 model is better performance than the others are.

7.2 Future Work

Based on the experimental results and from the point view of this research the following gaps are identified as prospective future research directions:

Future work in this area should include other components like ear, eyebrows, chin and forehead. It would be worthwhile to conduct further experiments with these face components to devise a way which can improve face recognition performance. In addition, increase the size of the database helps to enhance the recognition accuracy and the database should consider Partial face image (masks, hats, sunglasses) and Mixed variations (Pose + Illumination, Pose + Expression, Pose + Illumination + Expression).

In its current form, the developed model can be applied in different application scenarios. However, to develop full-face recognition system that applies for all applications, it is necessary to include faces that are inclined down. This paper directs to build a face recognition model that includes inclined, partial face image and mixed variations.

References

- [1] R. S. Kute, V. Vyas, and A. Anuse, "Component-based face recognition under transfer learning for forensic applications," *Inf. Sci. (Ny)*, vol. 476, pp. 176–191, 2019, doi: 10.1016/j.ins.2018.10.014.
- [2] H. F. Neo, C. C. Teo, and A. B. J. Teoh, "Development of partial face recognition framework," *Proc. - 2010 7th Int. Conf. Comput. Graph. Imaging Vis. CGIV 2010*, pp. 142–146, 2010, doi: 10.1109/CGIV.2010.29.
- [3] K. Sato, S. Shah, and J. K. Aggarwal, "Partial face recognition using radial basis function networks," *Proc. - 3rd IEEE Int. Conf. Autom. Face Gesture Recognition, FG 1998*, pp. 288–293, 1998, doi: 10.1109/AFGR.1998.670963.
- [4] F. Becerra-Riera, A. Morales-González, and H. Méndez-Vázquez, "Facial marks for improving face recognition," *Pattern Recognit. Lett.*, vol. 113, pp. 3–9, 2018, doi: 10.1016/j.patrec.2017.05.005.
- [5] K. Bonnen, B. F. Klare, and A. K. Jain, "Component-based representation in automated face recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 239–253, 2013, doi: 10.1109/TIFS.2012.2226580.
- [6] A. Vijayalakshmi and P. Raj, "An approach to improvise recognition rate from occluded and pose variant faces," *4th IEEE Spons. Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2015*, pp. 547–552, 2015, doi: 10.1109/ICCPEIC.2015.7259514.
- [7] S. Zhang and R. He, "Beyond Face Rotation : Global and Local Perception GAN for Photorealistic and," *Iccv*, pp. 2439–2448, 2017.
- [8] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1283–1292, 2017, doi: 10.1109/CVPR.2017.141.
- [9] C. Sagonas, Y. Panagakis, S. Zafeiriou, and M. Pantic, "Robust statistical face frontalization," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 3871–3879, 2015, doi: 10.1109/ICCV.2015.441.
- [10] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Towards Large-Pose Face Frontalization in the Wild Supplementary Materials," pp. 1–6.
- [11] F. Becerra-riera and A. Morales-gonz, "Detection and matching of facial marks in face

- images,” *Rev. Cuba. Ciencias Informáticas*, vol. 10, no. 1, pp. 172–181, 2016.
- [12] U. Park and A. K. Jain, “Face matching and retrieval using soft biometrics,” *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 3, pp. 406–415, 2010, doi: 10.1109/TIFS.2010.2049842.
- [13] S. A. Padmanabhan, “An efficient face recognition system based on hybrid optimized KELM
Content courtesy of Springer Nature , terms of use apply . Rights reserved . Content courtesy of Springer Nature , terms of use apply . Rights reserved .,” pp. 10677–10697, 2020.
- [14] S. Dwivedi and N. Gupta, “A New Hybrid Approach on Face Detection and Recognition,” vol. 3, pp. 485–492, 2020, doi: 10.31219/osf.io/r7984.
- [15] S. I. Ch’ng, K. P. Seng, and L. M. Ang, “Multiview-multiband face recognition system to solve illumination and pose variation,” *Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2010*, vol. 5, pp. 384–388, 2010, doi: 10.1109/ICCSIT.2010.5564099.
- [16] H. Mesfin, “Face Recognition using Principle Component Analysis,” Adama Science and Technology University, online Available : <http://localhost:8080/xmlui/handle/123456789/768>.
- [17] J. Ashok and V. Shivashankar, “An overview of biometrics,” *Acta Tech. Napocensis*, vol. 02, no. 07, pp. 2402–2408, 2006, [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:An+overview+of+biometric+s#0>.
- [18] A. K. Jain and A. Ross, “Bridging the gap: From biometrics to forensics,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 370, no. 1674, 2015, doi: 10.1098/rstb.2014.0254.
- [19] J. Fagertun, “Face recognition,” Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2005., 2005.
- [20] Y. Guo, S. Ruan, J. Landre, and J.-M. Constans, “Robust Face Recognition,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 7, pp. 1620–7, 2010, [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/20483699>.
- [21] A. Faizi, “Robust Face Detection Using Template Matching Algorithm,” *J. Orthop. Traumatol.*, vol. 9, no. Supplement 1, pp. 1–94, 2008, [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19541046>.
- [22] S. Karamizadeh, S. M. Abdullah, and M. Zamani, “An Overview of Holistic Face Recognition,” vol. 2, no. 9, pp. 738–741, 2013.

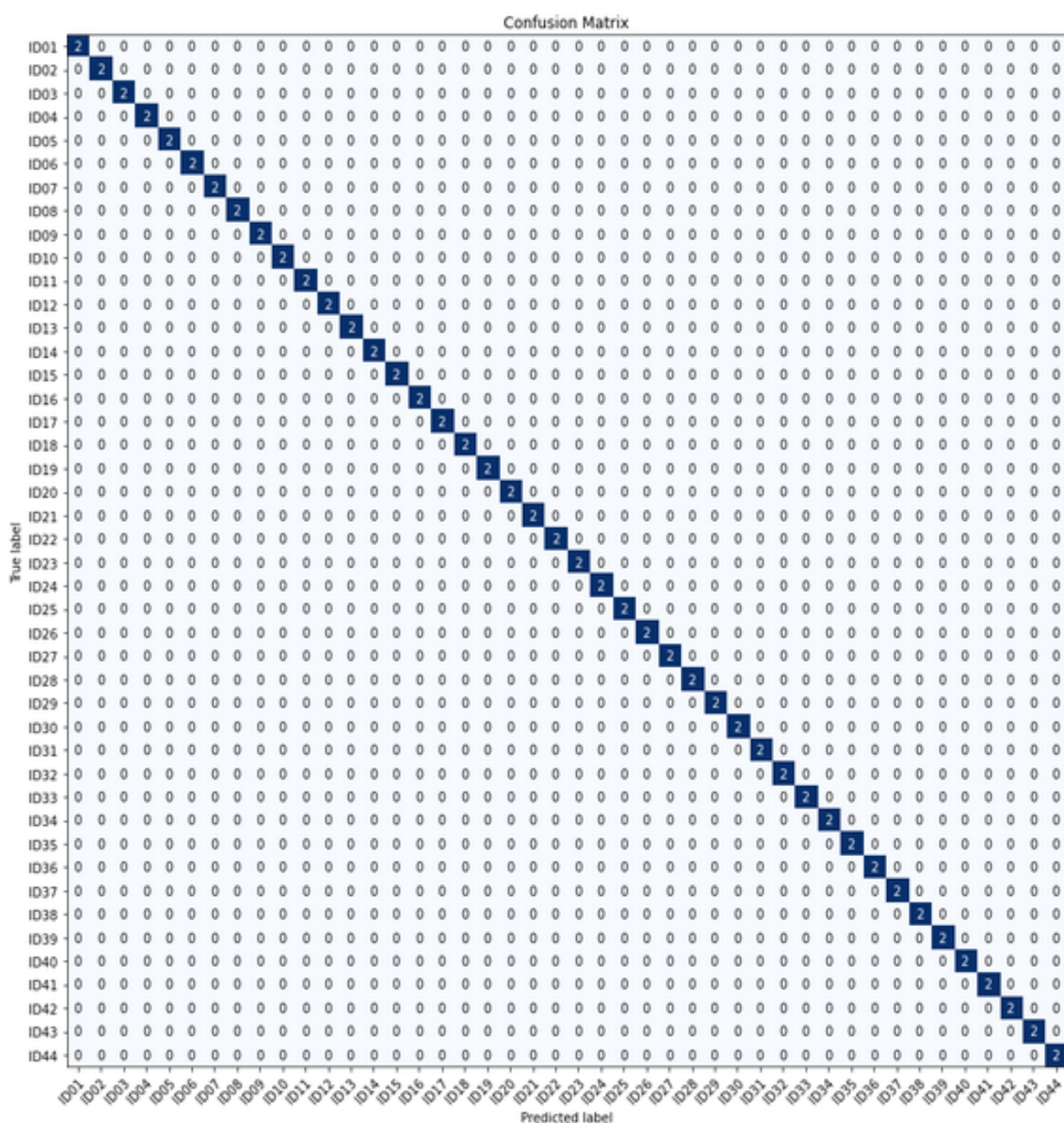
- [23] B. L. Priya and D. M. P. R. Rani, "Face Recognition System Techniques and Approaches," *Indian J. Appl. Res.*, vol. 4, no. 4, pp. 109–113, 2011, doi: 10.15373/2249555x/apr2014/32.
- [24] U. Park and U. Park, "Face Recognition : face in video , age invariance , and facial marks By Submitted to By," *Michigan State Univ. Dep. Comput.*, 2009.
- [25] M. Sharif, M. A. Ali, M. Raza, and S. Mohsin, "Face Recognition Using Edge Information and DCT," *Sindh Univ. Res. J. - SURJ (Science Ser.)*, vol. 43, no. 2, pp. 209–214, 2011, [Online]. Available: <http://sujo.usindh.edu.pk/index.php/SURJ/article/view/1371>.
- [26] T. Author, I. Date, T. Version, and U. R. L. Doi, "Osaka University Knowledge Archive : OUKA," no. Report I, 2018, [Online]. Available: <https://ci.nii.ac.jp/naid/120006484589>.
- [27] V. Contreras-gonzalez and V. H. Diaz-ramirez, "Facial landmark detection and tracking with dynamically adaptive matched filters," vol. 29, no. 3, 2020, doi: 10.1117/1.JEI.29.3.033004.
- [28] Z. H. Choudhury and K. M. Mehata, "Robust Facial Marks Detection Method Using AAM And SURF," vol. 2, no. 6, pp. 708–715, 2012.
- [29] G. Guo and N. Zhang, "A survey on deep learning based face recognition ☆," *Comput. Vis. Image Underst.*, vol. 189, no. July, p. 102805, 2019, doi: 10.1016/j.cviu.2019.102805.
- [30] S. Almabdy and L. Elrefaei, "Deep convolutional neural network-based approaches for face recognition," *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204397.
- [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [32] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *arXiv*, no. 1, 2018.
- [33] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-mobilenet models," *Mob. Inf. Syst.*, vol. 2020, 2020, doi: 10.1155/2020/7602384.
- [34] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep Face Recognition: A Survey," *Proc. - 31st Conf. Graph. Patterns Images, SIBGRAPI 2018*, pp. 471–478, 2019, doi: 10.1109/SIBGRAPI.2018.00067.
- [35] M. Usman, S. Latif, and J. Qadir, "Using deep autoencoders for facial expression recognition," *Proc. - 2017 13th Int. Conf. Emerg. Technol. ICET2017*, vol. 2018-Janua, pp. 1–6, 2018, doi:

- 10.1109/ICET.2017.8281753.
- [36] P. Tome, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Facial soft biometric features for forensic face recognition," *Forensic Sci. Int.*, vol. 257, pp. 271–284, 2015, doi: 10.1016/j.forsciint.2015.09.002.
 - [37] A. G. Tarres, F.; Rama, "GTAV Database." <http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.htm>.
 - [38] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.
 - [39] E. Culurciello, "Neural Network Architecture," 2015, [Online]. Available: <https://towardsdatascience.com/neural-network-architectures-156e5bad51ba>.
 - [40] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, no. October, 2017.
 - [41] H. Lamba, "One Shot Learning with Siamese Networks using Keras," [Online]. Available: <https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d>.
 - [42] "Introduction to OpenCV-Python Tutorials," [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_intro/py_intro.html.
 - [43] Jeff, "deep learning framework," [Online]. Available: <https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a>.
 - [44] C. Science and A. Science, "Object Detection for Mobile Robot Navigation in Dynamic Indoor Environment Object Detection for Mobile Robot Navigation in Dynamic Indoor Environment," no. September, 2019.
 - [45] J. Brownlee, "MTCNN," 2019. <https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/>.
 - [46] F. Recognition, *Handbook of Face Recognition*. DOI: 10.1007/978-0-85729-932-1, ISBN: 9780857299314, 2011.
 - [47] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

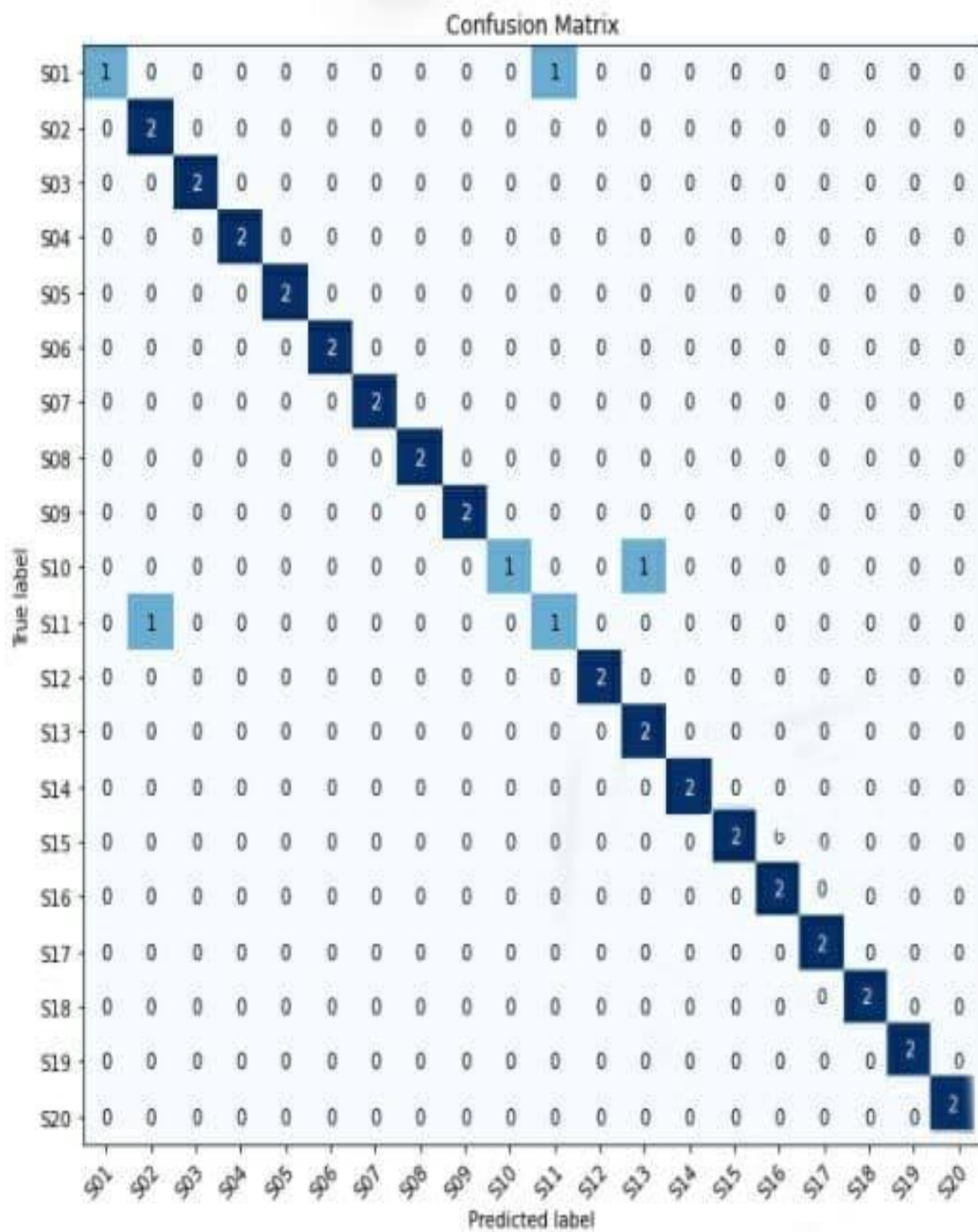
- [48] J. Brownlee, “How to Configure Image Data Augmentation in Keras.” <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>.
- [49] R. Gradilla, “Multi-task Cascaded Convolutional Networks (MTCNN) for Face Detection and Facial Landmark Alignment.” <https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923#:~:text=Multi-task Cascaded,eyes%2C nose%2C and mouth.>
- [50] Y. Zhong, J. Chen, and B. Huang, “Toward End-to-End Face Recognition Through Alignment Learning,” *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1213–1217, 2017, doi: 10.1109/LSP.2017.2715076.
- [51] H. R. Chou, J. H. Lee, Y. M. Chan, and C. S. Chen, “Data-Specific Adaptive Threshold for Face Recognition and Authentication,” *Proc. - 2nd Int. Conf. Multimed. Inf. Process. Retrieval, MIPR 2019*, no. October 2018, pp. 153–156, 2019, doi: 10.1109/MIPR.2019.00034.
- [52] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” *arXiv*, pp. 4510–4520, 2018.

Appendixes

Appendix I. Confusion Matrix for Feature Extractor and One-shot Model



Appendix II. Confusion Matrix for One-shot Model for Self-prepared Dataset



Appendix III. Evaluation Confusion Matrix Implementation

```
1 def plot_confusion_matrix(cm, classes, title, normalize=False, cmap=plt.cm.Blues):
2
3     fig, ax = plt.subplots(figsize=(15, 15))
4     im = ax.imshow(cm, interpolation='nearest', cmap=cmap)
5     ax.figure.colorbar(im, ax=ax)
6     # We want to show all ticks...
7     ax.set(xticks=np.arange(cm.shape[1]),
8           yticks=np.arange(cm.shape[0]),
9           # ... and label them with the respective list entries
10          xticklabels=classes, yticklabels=classes,
11          title=title,
12          ylabel='True label',
13          xlabel='Predicted label')
14
15     # Rotate the tick labels and set their alignment.
16     plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
17            rotation_mode="anchor")
18
19     # Loop over data dimensions and create text annotations.
20     fmt = '.2f' if normalize else 'd'
21     thresh = cm.max() / 2.
22     for i in range(cm.shape[0]):
23         for j in range(cm.shape[1]):
24             ax.text(j, i, format(cm[i, j], fmt),
25                   ha="center", va="center",
26                   color="white" if cm[i, j] > thresh else "black")
27     fig.tight_layout()
28     return ax
```

```
1 # Recall = TP / (TP + FN)
2 yishak_recall = np.diag(confMat) / np.sum(confMat, axis=1)
3
4 # Precision = TP / (TP + FP)
5 yishak_precision = np.diag(confMat) / np.sum(confMat, axis=0)
6 h = 0
7 for i in range(44):
8     h = h + confMat[i][i]
9 acc = h/len(all_labels)
10
11 yishak_f1 = (2 * yishak_recall * yishak_precision) / (yishak_recall + yishak_precision)
12
13 print("Accu = " + str(np.mean(acc)))
14 print("Recall = " + str(np.mean(yishak_recall)))
15 print("Precision = " + str(np.mean(yishak_precision)))
16 print("F1 = " + str(np.mean(yishak_f1)))
```

Appendix IV. Data Loader and Generator Implementation

```
1 # load data
2 def load_data(dataset):
3     print(dataset+"-----")
4     datas = []
5     data = []
6     labels = []
7     l = -1
8     alldirnames = []
9     k = 0
10
11     for (dirpath, dirnames, filenames) in walk(dataset):
12         datas.extend(filenames)
13         if k == 0:
14             alldirnames.extend(dirnames)
15             k = k + 1
16     u = 0
17     alldirnames.sort()
18     print(str(len(alldirnames))+" Classes")
19     for i in range(len(alldirnames)):
20         for filename in glob.glob(dataset+"/"+alldirnames[i] + '/*.jpg'):
21             filename = filename.replace("\\", "/")
22             data.append(filename)
23             #print(filename)
24             labels.append(i)
25             u = u + 1
26
27     print(alldirnames[i]+" : "+str(u))
28     u = 0
29
30     return data, labels
```

```
1 # make pairs
2 def make_pairs_train(xs, ys):
3
4     num_classes = max(ys) + 1
5
6     digit_indices = [np.where(ys == i)[0] for i in range(num_classes)]
7     print(num_classes)
8
9     ImageX1 = []
10    ImageX1_land = []
11
12    ImageX2 = []
13    ImageX2_land = []
14
15    labels = []
16    batch_counter = 0
17
18    ...
```

```

17 |
18 |     yesT = 0
19 |     noT = 0
20 |     while (batch_counter < len(xs)):
21 |         #rand_ind = np.random.randint(len(xs))
22 |
23 |         # add a matching example
24 |         x1 = xs[batch_counter]
25 |         label1 = ys[batch_counter]
26 |         idx2 = random.choice(digit_indices[label1])
27 |
28 |         x2 = xs[idx2]
29 |         #print(len(ImageX1))
30 |         image = cv.imread (x1)
31 |         image1 = cv.resize(image, (224,224))
32 |
33 |         image2 = cv.imread (x2)
34 |         image3 = cv.resize(image2, (224,224))
35 |
36 |         faces = detector.detect_faces(image1)
37 |
38 |         cho = 0
39 |         # Draw a rectangle around the faces
40 |         for result in faces:
41 |             #print("face First")
42 |             # get coordinates
43 |             x, y, w, h = result['box']
44 |             #cv2.rectangle(image, (x-10, y-15), (x+w+15, y+h+15), (0, 255, 0), 2)
45 |             image = image1[y:y+h, x:x+w]
46 |
47 |             p = 0
48 |             for key, (g,h) in result['keypoints'].items():
49 |                 theImage = image1[h-20:h+20, g-20:g+20]
50 |                 #print(image.shape)
51 |                 if(theImage.shape[0] > 1 and theImage.shape[1] > 1):
52 |                     #print("cool")
53 |                     theImage = cv.resize(theImage, (56,56))
54 |                     if(p == 0):
55 |                         im_h = theImage
56 |                         p = p + 1
57 |                         #print("p = 1")
58 |                     else:
59 |                         im_h = cv2.addWeighted(im_h,0.5,theImage,0.5, 0)
60 |                         #print("img_h")
61 |
62 |             if(p>0 and (image.shape[0] > 1 and image.shape[1] > 1)):
63 |
64 |                 ImageX1_land.append(im_h)
65 |                 ImageX1.append(image1)

```

```

66         cho = 1
67
68     if(cho == 1):
69
70         faces = detector.detect_faces(image3)
71         # Draw a rectangle around the faces
72         for result in faces:
73
74             # get coordinates
75             x, y, w, h = result['box']
76             #cv2.rectangle(image, (x-10, y-15), (x+w+15, y+h+15), (0, 255, 0), 2)
77             image = image3[y:y+h, x:x+w]
78             p = 0
79             for key, (g,h) in result['keypoints'].items():
80                 theImage = image3[h-20:h+20, g-20:g+20]
81                 if(theImage.shape[0] > 1 and theImage.shape[1] > 1):
82                     theImage = cv.resize(theImage, (56,56))
83                     if(p == 0):
84                         im_h = theImage
85                         p += 1
86                     else:
87                         im_h = cv2.addWeighted(im_h,0.5,theImage,0.5, 0)
88
89             if(p>0 and (image.shape[0] > 1 and image.shape[1] > 1)):
90                 ImageX2_land.append(im_h)
91                 ImageX2.append(image3)
92                 #print("Right: True")
93                 labels.append(1)
94                 yesT = yesT + 1
95             else:
96                 ImageX1_land.pop()
97                 ImageX1.pop()

```

```

99     cho2 = 0
100     # add a not matching example
101     label2 = random.randint(0, num_classes-1)
102     while label2 == label1:
103         label2 = random.randint(0, num_classes-1)
104
105     idx2 = random.choice(digit_indices[label2])
106     x2 = xs[idx2]
107
108     image = cv.imread (x1)
109     image1 = cv.resize(image, (224,224))
110     image2 = cv.imread (x2)
111     image3 = cv.resize(image2, (224,224))
112
113     faces = detector.detect_faces(image1)
114     # Draw a rectangle around the faces
115     for result in faces:
116         # get coordinates
117         x, y, w, h = result['box']
118         #cv2.rectangle(image, (x-10, y-15), (x+w+15, y+h+15), (0, 255, 0),
119         image = image1[y:y+h, x:x+w]
120
121     p = 0
122     for key, (g,h) in result['keypoints'].items():
123         #print("Cord")
124         theImage = image1[h-20:h+20, g-20:g+20]
125         if(theImage.shape[0] > 1 and theImage.shape[1] > 1):
126             theImage = cv.resize(theImage, (56,56))
127             if(p == 0):
128                 im_h = theImage
129                 p += 1
130             else:
131                 im_h = cv2.addWeighted(im_h,0.5,theImage,0.5, 0)
132
133     if(p>0 and (image.shape[0] > 1 and image.shape[1] > 1)):
134         ImageX1_land.append(im_h)
135         image = cv.resize(image, (224,224))
136         ImageX1.append(image1)
137         cho2 = 1
138         #print("Left: False")
139     if(cho2 == 1):
140         faces = detector.detect_faces(image3)
141         # Draw a rectangle around the faces
142         for result in faces:
143             # get coordinates
144             x, y, w, h = result['box']
145             #cv2.rectangle(image, (x-10, y-15), (x+w+15, y+h+15), (0, 255, 0)
146             image = image3[y:y+h, x:x+w]

```

```

148         p = 0
149         for key, (g,h) in result['keypoints'].items():
150             theImage = image3[h-20:h+20, g-20:g+20]
151             if(theImage.shape[0] > 1 and theImage.shape[1] > 1):
152                 theImage = cv.resize(theImage, (56,56))
153                 if(p == 0):
154                     im_h = theImage
155                     p += 1
156                 else:
157                     im_h = cv2.addWeighted(im_h,0.5,theImage,0.5, 0)
158
159             if(p>0 and (image.shape[0] > 1 and image.shape[1] > 1)):
160                 ImageX2_land.append(im_h)
161                 ImageX2.append(image3)
162                 labels.append(0)
163                 noT = noT + 1

```

```

164         else:
165             ImageX1_land.pop()
166             ImageX1.pop()
167
168         print(batch_counter)
169         batch_counter = batch_counter + 1
170         # plt.imshow(cv2.cvtColor(x_left_land[0], cv2.COLOR_BGR2RGB))
171         return ImageX1, ImageX1_land, ImageX2, ImageX2_land, labels
172
173

```

```

1
2 def myBatchGenerator(batch_size, lImages, lComp, rImages, rComp, labels):
3     while True:
4         batch_counter = 0
5
6         batch_imageL = []
7         batch_compL = []
8         batch_imageR = []
9         batch_compR = []
10        batch_label = []
11        while batch_counter < batch_size:
12            rand_ind = np.random.randint(len(lImages))
13
14            batch_imageL.append(lImages[rand_ind])
15            batch_compL.append(lComp[rand_ind])
16            batch_imageR.append(rImages[rand_ind])
17            batch_compR.append(rComp[rand_ind])
18            batch_label.append(labels[rand_ind])
19
20            batch_counter = batch_counter + 1
21
22        BIL = np.array(batch_imageL)
23        BILC = np.array(batch_compL)
24        BIR = np.array(batch_imageR)
25        BIRC = np.array(batch_compR)
26        BL = np.array(batch_label)
27
28        yield [BIL, BILC, BIR, BIRC], BL

```

Appendix V. Training Accuracy and Loss Plot Implementation

```

1 # list all data in history
2 print(history.history.keys())
3 # summarize history for accuracy
4 plt.plot(history.history['binary_accuracy'])
5 plt.plot(history.history['val_binary_accuracy'])
6 plt.title('model accuracy')
7 plt.ylabel('accuracy')
8 plt.xlabel('epoch')
9 plt.legend(['train', 'test'], loc='upper left')
10 plt.show()
11 # summarize history for loss
12 plt.plot(history.history['loss'])
13 plt.plot(history.history['val_loss'])
14 plt.title('model loss')
15 plt.ylabel('loss')
16 plt.xlabel('epoch')
17 plt.legend(['train', 'test'], loc='upper left')
18 plt.show()

```