FACE RECOGNITION UNDER PARTIAL OCCLUSION: A DETECTION AND EXCLUSION OF OCCLUDED FACE REGIONS APPROACH

JUDITH NYABOKE ABIERO

MASTER OF SCIENCE

(Software Engineering)

JOMO KENYATTA UNIVERSITY

OF

AGRICULTURE AND TECHNOLOGY

2023

Face Recognition under Partial Occlusion: A Detection and Exclusion of Occluded Face Regions Approach

Judith Nyaboke Abiero

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Software Engineering of the Jomo Kenyatta University of Agriculture and Technology

2023

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university

Signature…….Date……

Judith Nyaboke Abiero

This thesis has been submitted with our approval as the university supervisors

Signature…….Date……

Dr. Michael W. Kimwele, PhD

JKUAT, Kenya

Signature…….Date……

Dr. Geoffrey W. Chemwa, PhD

JKUAT, Kenya

DEDICATION

I dedicate this research to the giver of life, my family, my friends, my supervisors, my university and everyone else who directly or indirectly helped in this research.

ACKNOWLEDGEMENT

I acknowledge and appreciate my family, research supervisors; Dr. Kimwele and Dr. Chemwa, Mr. Sylvester Kiptoo and friends whose dedication and support has helped me a lot in completion of the research thesis.

I also acknowledge those who directly or indirectly helped me complete this research thesis.

TABLE OF CONTENTS

DECLARATION ..... **ii**

DEDICATION ..... **iii**

ACKNOWLEDGEMENT ..... **iv**

LIST OF TABLES ..... **ix**

LIST OF FIGURES ..... **x**

LIST OF APPENDICES ..... **xiii**

LIST OF ABBREVIATIONS/ ACRONYMS ..... **xiv**

ABSTRACT ..... **xvi**

CHAPTER ONE **1**

INTRODUCTION **1**

1.1 Background 1

1.2 Problem Statement 3

1.3 Research Questions 4

1.4 Research Objectives 4

1.4.1 Specific objectives 4

1.5 Scope of the Research 5

1.6 Thesis Organization 5

CHAPTER TWO 6

LITERATURE REVIEW 6

2.1 Introduction 6

2.2 Face Detection and Recognition 6

2.2.1 Face Detection 6

2.2.2 Face Recognition 10

2.3 Partial Occlusion in Face Recognition 18

2.4 Problems with Partial Occlusions in Face Recognition 19

2.5 Face Recognition Approaches under Partial Occlusions 20

2.5.1 Occlusion Scenarios 20

2.5.2 Occlusion Robust Approaches under Face Recognition 20

2.6 Summary 32

2.7 Conceptual Model 34

CHAPTER THREE 36

RESEARCH METHODOLOGY 36

3.1 Introduction 36

3.2 The Proposed Approach 36

3.2.1 The proposed detection and exclusion of occluded face regions approach algorithm 43

3.3 Data Collection 44

3.4 Experimental Setup 45

3.4.1 System Specifications 46

3.4.2 Development Tools 46

3.4.3 Data Pre-processing 46

3.4.4 Selecting the Data Sample 46

3.4.5 Face Alignment 47

3.4.6 Masking Faces from the Pubfig, FaceScrub and Yale B Datasets 48

3.4.7 Data Augmentation 49

3.4.8 Feature Extraction 50

3.4.9 Feature Normalization and Dimensionality Reduction 52

3.4.10 Classifier Training 53

3.4.11 Trained Classifier/ Model Evaluation 54

3.5 Ethical Considerations 56

3.6 Conclusion 56

CHAPTER FOUR 57

EXPERIMENTS' RESULTS ANALYSIS AND DISCUSSION 57

4.1 Introduction 57

4.2 Experiments' Results and Analysis 57

4.2.1 Non-Occluded Region Retrieval using Skin Detection 57

4.2.2 Non-Occluded Region Retrieval using Haar Cascade Classifiers 60

4.2.3 Performance on the LFW Dataset 63

4.2.4 Recognition Rates under Different Block Occlusion Coverage Area 64

4.3 Discussion 66

4.3.1 Summary of Key Findings 67

4.3.2 Comparison of Results with those of Existing Approaches 69

4.4 Effect of the Number of Images in Learning a Class 72

4.5 Limitations of the Research 73

4.6 Research Outputs 74

4.7 Summary 74

CHAPTER FIVE 75

SUMMARY, CONCLUSIONS AND FUTURE WORK 75

5.1 Introduction 75

5.2 Research Summary 75

5.3 Conclusions 76

5.4 Future Work 76

REFERENCES 77

APPENDICES 87

LIST OF TABLES

Table 2.1: A categorization of occlusion challenges 19

Table 2.2: Summary of occlusion approaches' strengths and weaknesses 32

Table 3.1: Summary of Data Collected 45

Table 4.1: Recognition rates using resnet18 and skin detection 58

Table 4.2: Recognition rates using vgg11 and skin detection 59

Table 4.3: Recognition rates using Inception Resnet (V1) and skin detection 60

Table 4.4: Recognition rates using resnet18 and haar cascade classifiers 61

Table 4.5: Recognition rates using vgg11 and haar cascade classifiers 62

Table 4.6: Recognition rates using Inception Resnet (V1) and haar cascade classifiers 63

Table 4.7: Performance of the detection and exclusion of occluded face regions approach 69

Table 4.8: Performance comparison of various approaches to ours 70

LIST OF FIGURES

Figure 1.1: Example of an un-occluded face image from the Webface-OCC dataset) 2

Figure 1.2: Examples of partially occluded face images from the Webface-OCC dataset) 2

Figure 2.1: Face recognition process flow. 6

Figure 2.2: Diverse dimensions of Haar features for Viola-Jones face detection. 8

Figure 2.3: Cascade classifier stages..... 9

Figure 2.4: Principal component analysis problem formulation. Retrieved from ... 11

Figure 2.5: Representation of the model-based method. 13

Figure 2.6: Illustration of the three main layers of an MLP 14

Figure 2.7: Representation of a full convolution neural network. 16

Figure 2.8: Evolution of face recognition methods 17

Figure 2.9: Representation of an occluded face sample as the linear combination of training sample with some intra-class variations plus the error. 22

Figure 2.10: Representation of the occluded image *y*, by a linear combination of all training data in the dictionary *A* and added by a residual image *x* standing for occlusion whereby *L* is the residue 25

Figure 2.11: Flowchart of the joint and collaborative representation with local adaptive feature model 26

Figure 2.12: Deep feature vectors represented visually from 5 subjects from the AR dataset. 27

Figure 2.13: The overview of the PDSN framework where b_i b_j the non-overlapping face blocks, M_i and M_j binarized feature discarding mask. 30

Figure 2.14: The pairwise differential siamese network. 31

Figure 2.15: An illustration of the alignment free approach). 32

Figure 2.16: Conceptual model of the proposed approach 34

Figure 3.1: Representation of the approach process flow 37

Figure 3.2: Representation of the face as a Cartesian plane 38

Figure 3.3: An example image from the Pubfig dataset used to demonstrate equation (1) to (4) 39

Figure 3.4: Deriving O_u,O_d,O_l,O_r from the occluded face 40

Figure 3.5: Deriving Ou, Od, Ol, Or from a synthetically occluded image 40

Figure 3.6: A skin segmented face image 42

Figure 3.7: The least occluded region selected from Figure 3.5 42

Figure 3.8: Face that is not masked retrieved from the Facescrub dataset 49

Figure 3.9: A synthetically masked face 49

Figure 3.10: Original face image from FaceScrub dataset with its augmentations . 50

Figure 3.11: The vgg 11 input, layers and output 51

Figure 3.12: Resnet-18 architecture 52

Figure 3.13: Schema for Inception-Resnet-v1 and Inception-ResNet-v2 networks . 52

Figure 3.14: Training phase flowchart 53

Figure 3.15: Testing Phase flowchart 55

Figure 4.1: Recognition rates with different block occlusion percentages on the FaceScrub dataset 64

Figure 4.2: A face image retrieved from the FaceScrub dataset with 80% block occlusion..... 64

Figure 4.3: Performance comparison among different least occluded face sections 65

Figure 4.4: The detection and exclusion of occluded face regions approach 68

Figure 4.6: ROC curves from the Pubfig dataset (Kumar et al., 2009) our experiments 72

Figure 4.5: ROC curves from the Pubfig dataset 72

Figure 4.7: Chart showing the effect of number of images on recognition rate 73

LIST OF APPENDICES

Appendix I: The Graphical User Interface 87

Appendix II: Sample codes..... 90

LIST OF ABBREVIATIONS/ ACRONYMS

CCTV- Closed-circuit Television

LGBP – Local gabor binary pattern

CNN – Convolutional neural network

LBP – Local binary patterns

SIFT – Scale invariant feature transform

HOG- Histogram of oriented gradients

PCA – Principal component analysis

LDA – Linear discriminant analysis

DDRC- Deep dictionary representation-based classification

PDSN – Pairwise differential siamese network

SRC – Sparse representation classification

LFW- Labelled faces in the wild

Pubfig – Public Figures Face Dataset

Yale B - Extended Yale Face Dataset B

HSV – Hue, saturation and value

YCbCr – Luma, blue and red components

ResNet – Residual neural network

VGG – Visual geometry group

VGGFace – Visual geometry group Face

MLP- Mulitlayer perceptron

TP – True positive

TN – True negative

FN – False negative

FP – False positive

ROC – Receiver operating curve

ABSTRACT

Partial face occlusions such as scarfs, masks and sunglasses compromise face recognition accuracy. This thesis presents a face recognition approach robust to partial occlusions. The approach adopted for this study is based on the assumption that the human visual system ignores occlusion and solely focuses on the non-occluded sections for recognition. Four sections derived from a whole/ un-occluded image and the whole face are used to train a classifier for recognition. For testing, an occluded face image is also divided into the four sections above from which, the non-occluded or the least occluded section is selected for recognition. Two strategies were used for occlusion detection; skin detection and the use of haar cascade classifiers. This thesis mitigated weaknesses from literature review such as use of datasets that simulate real world occlusion scenarios, use of less data in training and not requiring any type of occlusion variation in training data. Additionally, the classifier performed relatively well in the classification task with an accuracy of 92% on the Webface-OCC, 96% on the Pubfig, 92% on the FaceScrub, 96% on the Yale B and 92% on the LFW datasets.

CHAPTER ONE

INTRODUCTION

1.1 Background

The face is a feature that best distinguishes a human being hence it can be crucial for human identification (Kakkar & Sharma, 2018). Face recognition is the ability to recognize human faces, this can be done by humans and advancements in computing have enabled similar recognitions to be done automatically by machines. The face recognition process involves three stages; face detection, feature extraction and classification and face recognition. Face detection determines whether a human face appears in a given image or not and where the faces are located. In feature extraction the human face patches are extracted from images (Bansal, 2018) whereas the face recognition phase involves determining the identities of the faces from which facial features had been extracted. A face database is required and, for each individual, several images are taken and their features extracted and stored in the database (Bansal, 2018).

Face recognition can be applied in many areas such as criminal investigations in the detection and identification of criminals from surveillance videos. The enhanced system should be able to detect an object, track an object, classify or identify an object and analyse its activity automatically (Chen et al., 2018). In modern closed-circuit television (CCTV) systems, criminals can be identified online as the systems are embedded with face recognition services (Xiao et al., 2019).

One of the challenges that face recognition systems face is partial occlusion; caused when some parts of the target image are not being obtained. This poses a challenge because facial recognition methods require the availability of a whole input face; partial features may lead to wrong classification (Satonkar et al., 2011). Examples of whole or non-occluded and occluded face images are shown in Figure 1.1 and Figure 1.2 respectively. Some of the occlusion robust approaches that have been used to counter the challenge are; extraction of local descriptors from the non-occluded

facial areas. Methods used in this approach are based on patch engineered features and learned features. The former involves hand crafted features like local binary patterns (LBP) or Gabor features, the latter include methods such as subspace learning, statistical learning sparse representation classifier and deep learning (Zhang et al., 2018). These methods have issues while using shallow features such as LGBP only in the hand-craft features (Song et al., 2019). Additionally, the need for face images to be aligned well so that features can be extracted hinders their application in real life (Zhang et al., 2018).

Figure 1.1: Example of an un-occluded face image from the Webface-OCC dataset (Huang et al, 2021)

Figure 1.2: Examples of partially occluded face images from the Webface-OCC dataset (Huang et al, 2021)

The second approach achieves this by recovering clean faces from the occluded faces. These methods use techniques such as reconstruction for face recognition or inpainting which considers the occluded face as a repair problem. This is done by sparse representation classification that makes use of dictionaries and sparse representations for classification. This approach's generalization is compromised because it requires identical samples of the testing and training sets. One of the most recent approaches is the use of convolutional neural networks (Song et al., 2019). The convolution neural networks (CNNs) have problems such as need of huge dataset for training, translation invariance and loss of valuable information through pooling layers (Tarrasse, 2018).

Occlusion aware face recognition methods assume that visible parts of the face are ready; therefore, during face recognition occluded parts are excluded. The methods used under this approach are either occlusion detection-based face recognition or partial face recognition. The former performs the occlusion detection first before obtaining a representation for face recognition from the un-occluded face parts only. The latter ignores the occlusion detection phase. Additionally, it is based on the assumption that a partial face is available and it is used for face recognition (Zhang et al., 2018).

Therefore, this research assumes that a partial face can be recognized from a set of both occluded faces with masks, sunglasses or other accessories and un-occluded faces by focusing on the un-occluded face features shown on the provided face. In the training phase, a whole/un-occluded face images are divided into four sections; vertically to produce the right and left sections of the face and horizontally to provide the upper and lower sections of the face. All these sections will be used in combination with whole faces are used to retrieve feature vectors from a pre-trained convolutional neural network (CNN) model that are used to train a classifier for face recognition. In the testing phase; the occluded face is divided into four sections as described above. Thereafter, occlusion detection is done on all the four regions and the least occluded region is selected and used for recognition. Such an algorithm can be used in criminal identification systems because; criminals have a tendency to hide part of their faces when committing crimes.

1.2 Problem Statement

The facial recognition methods require the accessibility of a whole input face, in the lack of the above it may lead to wrong grouping (Satonkar et al., 2011) or lead to a drop in recognition accuracy (Huang et al., 2021). Additionally, Cen and Wang (2019) also noted that recognizing partially occluded faces was error prone. According to (Min et al., 2014), face occlusions can be caused by reasons that can either be undeliberate or intentional. For example, in security, scarves, sunglasses and caps are worn by criminals to ensure their faces are not recognized while committing illegal crimes (Min et al., 2014). Additionally, facial expressions that are

exaggerated can also be regarded as occlusion (Wen et al., 2015). This in turn affects face recognition in that the discriminative facial features are distorted, therefore, the distance between two faces of the same image are increased and hence the intra class variations are larger than the inter-class variations (Satonkar et al., 2011). Additionally, occluded facial landmarks lead to registration errors, thus degrading the recognition rate. (Min et al., 2014) added that, in order to achieve robustness in face recognition, it is crucial to control partial occlusion.

In this research, a face recognition approach that is robust to partial occlusions such as masks, glasses and other face accessories was proposed. The approach learns features from the face provided; non-occluded faces and identifies a partially occluded face using the non-occluded section or part. Therefore, in real life applications it can enable identification of criminals whose faces are highly likely to be partially occluded in real time, whenever their face image is provided by investigating authorities.

1.3 Research Questions

i. What face recognition approaches robust to partial occlusions exist and what are their strengths and weaknesses?
ii. How can an enhanced facial recognition approach robust to partial occlusions be developed?
iii. How will the performance of the approach developed in (ii) above be evaluated?

1.4 Research Objectives

The main objective of this thesis was to develop an occlusion robust facial recognition approach derived from existing approaches.

1.4.1 Specific objectives

i. To analyse existing face recognition approaches that are robust to partial occlusions in order to identify their strengths and weaknesses.

ii. To develop a face recognition approach that is robust to partial occlusions derived from existing approaches.
iii. To evaluate the performance of the developed face recognition approach that is robust to partial occlusions.

1.5 Scope of the Research

This research aimed at developing a face recognition approach robust to partial occlusions. The justification of this research is that the proposed approach would use less computational cost, few training samples, learn more robust face features to enable face identification despite the existence of partial occlusions.

1.6 Thesis Organization

The thesis is organized and divided into five chapters. The first chapter introduces the research topic as the background of the study, introduces the problem statement, highlights the research objectives and research questions, the scope of the research and the thesis organization.

The second chapter of the thesis has the literature review that discusses the face recognition process and the methods used. It also discusses the partial occlusion problem in face recognition, the types of partial occlusions and approaches to mitigate the partial occlusion problem and their weaknesses. It also introduces the proposed approach robust to partial occlusion derived from the existing approaches.

The third chapter discusses the methodology used in this research. The proposed approach, the data collection process, the experimental setup and the ethical considerations. The fourth chapter introduces and discusses the results obtained from the experiments, limitations of the research and research outputs. The fifth chapter highlights the research summary, conclusions and future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In order to develop a robust face recognition approach in partial occlusion scenarios this thesis relied on the knowledge and understanding of face recognition in occlusion scenarios. The literature review covered here focused on occlusions, types of occlusions and approaches to mitigate partial occlusion in face recognition.

2.2 Face Detection and Recognition

Face detection involves searching for and locating human faces in images (Feng et al., 2022) whereas face recognition is the ability to identify the identity of a face through a given test image (Iliadis et al., 2017). A typical face recognition process is represented in Figure 2.1.

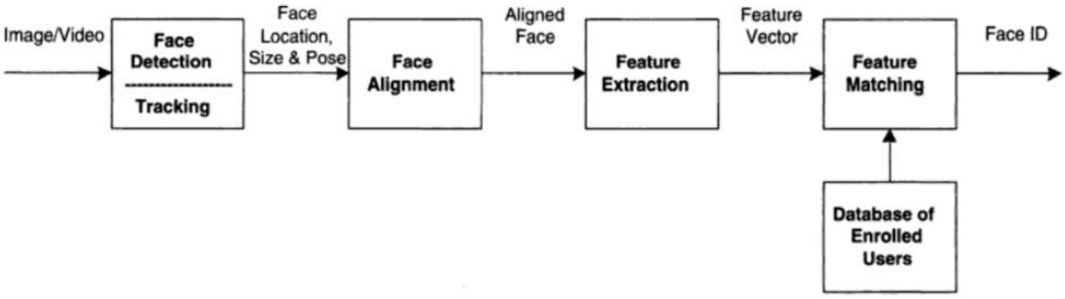


Figure 2.1: Face recognition process flow. Retrieved from (Kamalakumari & Vanitha,2016)

2.2.1 Face Detection

The aim of face detection has always been to identify if there is a face in a video or an image and if there is, return its location and extent of each face (Li et al., 2020). Some of the techniques of face detection include but are not limited to knowledge-based methods which are centred on geometry of the face and organization of the

face features. They describe the face's shape, size and texture or in other cases head, eyebrows, eyes, nose and chin (Kumar et al., 2017**).** Coming up with well-defined rules is a major challenge (Bernstein, 2020).

Feature invariant approaches which are used to catch physical features of a humanoid face despite the varying light settings; features used include but are not limited to skin colour, shape and texture. Such methods are very subtle to illumination, occlusion, existence of skin colour areas and neighbouring faces (Kumar et al.,2017). These features can be affected by light and noise negatively (Bernstein, 2020). Template-based methods can be considered to be sensitive to scale, pose and shape disparity of the human body whereas the deformable templates have been suggested to handle such variations (Kumar et al., 2017). Such methods do not address the pose, shape and scale variations (Bernstein, 2020).

Appearance based methods learn examples from examples in imageries. They mostly depend on statistical studies and machine learning techniques through which they can find features of face and a non-face that are relevant. The learned features are discriminant functions that can be used for face detection (Kumar et al., 2017). They can also be used for feature extraction (Bernstein, 2020). Some of the common face detection algorithms used include but are not limited to; viola jones, histogram of oriented gradients (HOG) and region-based convolution neural network (R-CNN).

The Viola and Jones (2003) face detection algorithm contains the following components; the Haar features, integral image, Adaboost classifier and a cascade structure. The Haar features are quadrilateral in shape as shown in Figure 2.2. The feature resultant of a single value is calculated by deducting the summation of pixels under white rectangles from the summation of pixels under black rectangles.

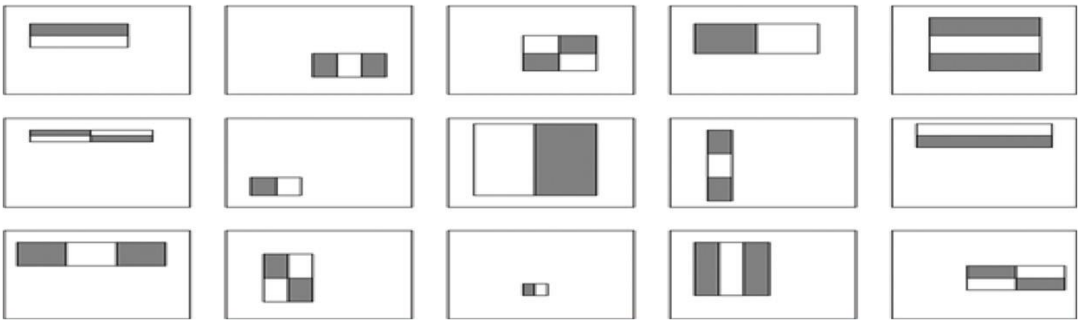


Figure 2.2: Diverse dimensions of Haar features for Viola-Jones face detection. Retrieved from (Mahdi et al., 2017)

The first step while using the Viola Jones algorithm is to find the integral image which is an algorithm for fast and proficiently computing the sum of values in a rectangular subset of a grid.

$$ii(x,y) = \sum_{x' \leqslant x, y' \leqslant y} i(x',y')$$

Whereby *ii(x,y)* is the integral image at pixel location *(x,y)* and the original image is *i(x'y')*.

Adaboost, a machine learning algorithm helps find the finest features among the many Haar like features extracted. After extraction, for it to be decided whether a window has a face or it doesn't, the weighted grouping of all the features is used in evaluation and decision making. A strong classifier is constructed by Adaboost through a linear combination of weak classifiers which would be the firstly acquired features. Thereafter, a cascade classifier is used, which contains stages, each with a strong classifier because a single strong classifier from a linear combination would be impractical to evaluate each window of an image due to computational cost. Each stage in the cascade is used to decide whether a specified sub-window is a face or not as shown in Figure 2.3. After the face has been detected it is saved for feature extraction.

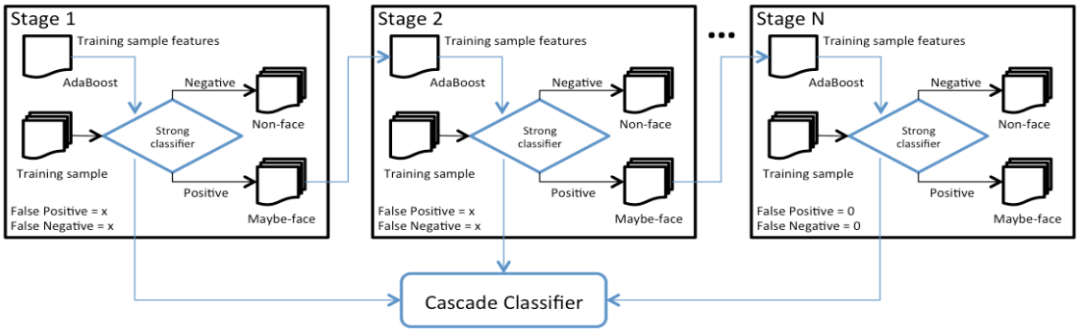


Figure 2.3: Cascade classifier stages. Retrieved from (Cen, N.D).

Histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for object detection. It focuses on the shape or structure of the image. It uses the magnitude and orientations of the gradients to generate histograms of the face image (Tyagi, 2021). According to (Saini, 2022), the HOG works by taking an input image M, analysing each pixel M(i) of image M for the relative dark pixels surrounding it directly. An arrow pointing in the direction of the flow of darkness relative to M(i) is added. The previous two steps are performed for each pixel. The arrows which equal to gradients replace each pixel. They show the flow from light to dark across an entire image. Since complex features like eyes can give too many gradients, the whole function, that is, the function that takes an input image M and replaces each pixel with an arrow or gradient is aggregated, to produce a global representation. Therefore, the image is broken up into 16 * 16 squares and assigned an aggregate gradient G' to each square. The function can be maximum of or minimum of (Saini, 2022).

Region-based convolution neural network (R-CNN) detects a face by generating region proposals on a CNN framework to localize and classify objects in images (Bernstein, 2020). It works by creating bounding boxes in regions using selective search (Saini, 2022). The selective search works by looking at an image through windows of different sizes. It then tries to group together adjacent pixels for each size by texture, colour or intensity to identify objects. According to (Saini, 2022), the R-CNN algorithm works by first generating regions for bounding boxes. The images in bounding boxes are run through a pre-trained neural network, to see what object is

in the box a support vector machine (SVM) is used. Finally, a linear regression model is used to output tighter coordinates for the box once the object has been classified.

2.2.2 Face Recognition

Face recognition is the ability to verify or visually identify a person using a face picture (Wanyonyi & Celik, 2022). Ideally, we need a system that is similar to a human eye in some sense to identify a person. With the advancement in technology this has been made possible by use of several approaches such as holistic matching, feature based methods, model-based methods, hybrid methods and neural networks.

Holistic matching is a method that takes a whole face region as input data into the face catching system (Saini, 2022). It is used to try to retrieve the most relevant features of a face (Dwivedi, 2018). Eigenfaces, PCA, Linear Discriminant Analysis and independent component analysis are examples of holistic models (Saini, 2022). Principal component analysis (PCA) developed by (Sirovich & Kirby,1986) is an unsupervised dimension reduction algorithm which simplifies the representation problem chosen for eigenvalues and corresponding eigen vectors to have a consistent representation. Its problem formulation is shown in Figure 2.4. Its advantages are; it deals with random noise; it reduces the distance between projection space and data space and it reduces redundancy. Its disadvantages are that it does not perform well rotation scaling translation distortions (Ismail & Sabri, 2009).

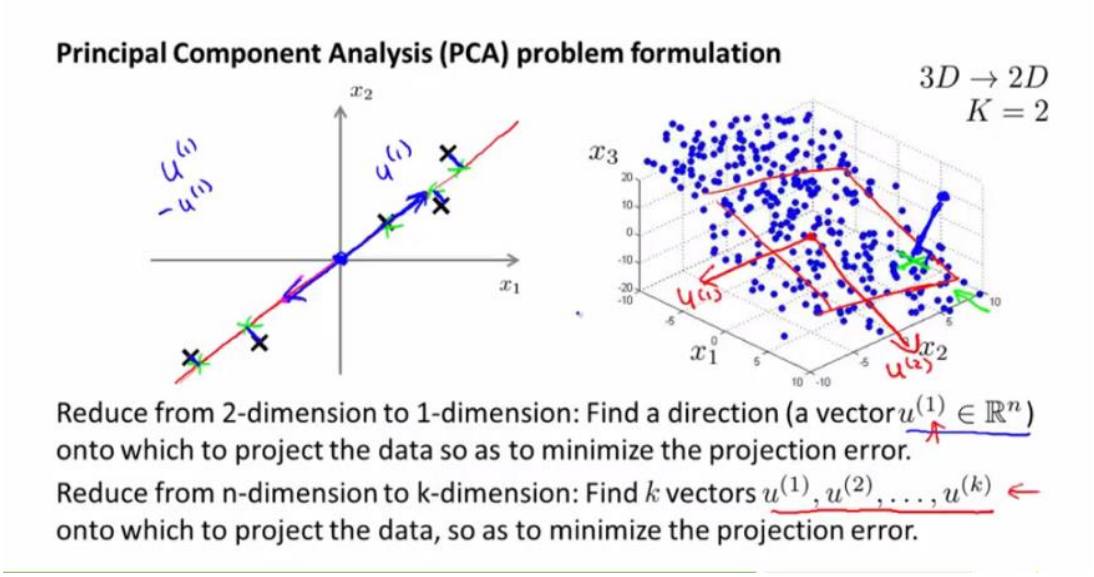


Figure 2.4: Principal component analysis problem formulation. Retrieved from (Dwivedi, 2018)

Eigenface designed by Turk and Pentland (1991), is a simple method that is less sensitive to pose variation and it also has better performance when small databases and training sets are used. It uses features such as eyes, mouth and nose on a face and relative distances between these features. In facial field these features are known as Eigen faces. It uses PCA a mathematical tool to extract facial features. The Eigen faces when combined can reconstruct an image from the training set (Ismail & Sabri, 2009).

According to Turk and Pentland (1991); given a set of m images of *N*N* dimension; convert the images into N^2 vectors $x_1, x_2, x_3 \ldots\ldots x_m$. Then calculate the average of all the face vectors $\psi = \frac{1}{m}\sum_{i=1}^{m} x_i$ and subtract it from each vector $a_i = x_i - \psi$. All the face vectors derive a matrix of size N^2 * M, $A = [a_1\ a_2\ a_3\ \ldots\ldots\ a_m]$. Then find the covariance matrix $Cov = A^T A$. The eigen values and eigenvectors of the above covariance matrix are calculated as; $A^T A v_i = \lambda_i v_i$, $AA^T A v_i = \lambda_i A v_i$, $C' u_i = \lambda_i u_i$. Where $C' = AA^T$ and $u_i = A v_i$.

Using the formula $u_i = A_{vi}$, the eigenvector and eigen value are calculated and mapped into *C'*. Then, select the K eigen vectors of *C'* that correspond to the K

largest value. Represent each face vectors in the linear combination of the best K eigenvectors $x_i - \psi = \sum_{j=1}^{K} w_j u_j$,where the u_j are the eignfaces. The training faces are represented in the from of vector of the coefficient of eigenfaces

$$x_i = \begin{bmatrix} w_1^i \\ w_2^i \\ w_3^i \\ . \\ . \\ w_k^i \end{bmatrix}.$$

For the testing phase; given an unknown face image y: have it centred have similar dimensions to the training image. Use $\phi = y - \psi$ to subtract the face from the average face ψ . Then obtain the linear combination of eigenfaces

$\phi = \sum_{i=1}^{k} w_i u_i$ by projecting the normailzed vector into eigenspace that generates the vector of the coefficient such that;

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ . \\ . \\ w_k \end{bmatrix}.$$

Take the above generated vector and subtract it from the training image to get the minimum distance between the training and testing vectors $e_r = min_l \, || \Omega - \Omega_l ||$. If e_r is less than the tolerance level T_n then the face is recognised, otherwise the face is not matched (Pawangfg, 2021).

On the other hand, Linear Discriminant Analysis (LDA) finds a linear combination of the features while it preserves class separability (Dwivedi, 2018). It can be used as a dimensionality reduction and classification method (Xiaozhou, 2020). It is usually used for supervised classification problems. In face recognition it reduces the number of features before the process of classification. For example; given a variable *V* that comes from one of *N* classes, having some class-specific probability densities *f(v);* a discriminant rule works by dividing the data space into *N* disjoint regions that represent all classes. For classification, therefore, the variable *v* has to be allocated to

class *j* if *v* is in region *j*. This can be done by either using the maximum likelihood or the Bayesian allocation rule (Xiazhou, 2020).

Feature based methods use local features such as nose, eyes and mouth. These features are first extracted and their locations, geometry and appearance are fed into a structural classifier. Some examples of methods under this are; generic methods based on edges, lines and curves, feature-template-based methods and structural matching methods (Saini, 2022).

Model based methods try to model a face (Saini, 2022). A face sample is introduced to a model whose parameters are used to recognise the image (Dwivedi, 2018). They can be classified into 2D and 3D (Saini, 2022). The model-based method is illustrated in Figure 2.5.

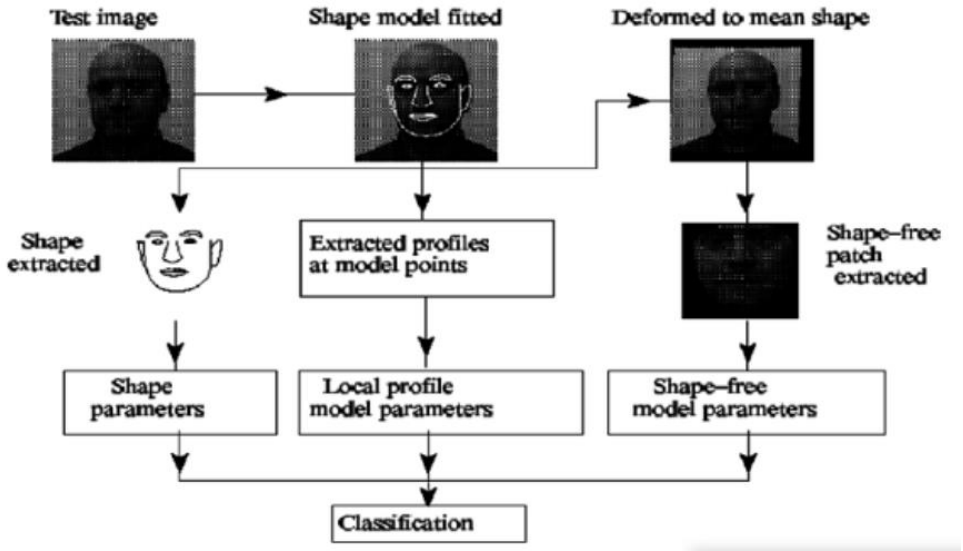


Figure 2.5: Representation of the model-based method. Retrieved from (Saini, 2022)

Hybrid methods combine both holistic and feature extraction methods (Saini, 2022). They use 3D images whereby; the curves of the eye sockets, shapes of the chin or forehead are noted. This 3D system includes detection, position, measurement, representation and matching.

Neural networks, on the other hand, simulate the human brain for face recognition (Li et al., 2020). For a neural network, a linear function and a nonlinear activation compose a neuron. A multilayer perceptron (MLP) is an example of a simple neural network. An MLP is a feed forward neural network that consists of three main layers; the input, hidden and output layers (Educative Answers Team, N/A) as shown in Figure 2.6.

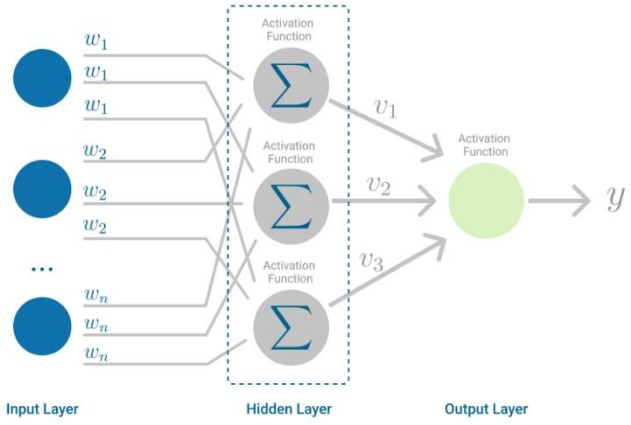


Figure 2.6: Illustration of the three main layers of an MLP. Retrieved from (Bento, 2021)

The input layer consists of neurons $\{x_i|x_1,x_2,x_3,\ldots\ldots,x_m\}$ that represent input features whereas the hidden layer takes values of inputs from the previous layer and transforms them with a weighted summation $w_1x_1+w_2x_2+\ldots\ldots+w_mx_m$ followed by a non-linear activation function $g(.):R \Rightarrow R$. Finally, the output layer transforms values of the last hidden layer into output values (Pedregosa et al., 2011).

The neurons are the building block of an MLP. Each neuron has a bias that has to be weighted. The weights are mostly initialized to small random values. These weighted values or inputs are summed and then passed through an activation function. An activation function can be described as mapping of summed weighted input into the output of the neuron. It also monitors the threshold at which the neuron is activated and the output signal's strength (Brownlee, 2016).

Deep learning is a machine learning algorithm that is built on the concept of the human brain and neurons' communication. The neurons in deep learning are virtual and they also perform statistical regressions in that they tend to learn data representations of several levels of feature extraction. Deep learning algorithms use very huge datasets of faces from which they learn rich and compact representations of faces. It allows better and faster identification. An example of a deep learning neural network is the convolutional neural network (CNN). A CNN is generally used as a feature extractor, followed by a classifier (Cen & Wang, 2019).

Convolutional neural networks are a variant of Multi-Layer Perception (MLP**),** inspired by the mammalian visual cortex of simple and intricate cells. Consisting of 4-8 layers with image processing tasks merged into the design, it applies three architectural concepts in its design; shared weights, local receptive field and subsampling (Syafeeza et al., 2014). A convolution function can be represented as;

$$s[t]=(x\star w)[t]=\sum_{a=-\infty}^{a=\infty} x[a]w[a+t]$$

Feature map Input kernel

by (Khandewal,2018).

It can also have an abstract description as;

$$x^1 \longrightarrow \boxed{w^1} \longrightarrow x^2 \longrightarrow \cdots \longrightarrow x^{L-1} \longrightarrow \boxed{w^{L-1}} \longrightarrow x^L \longrightarrow \boxed{w^L} \longrightarrow z$$

by (Wu, 2017), where *x* is the input, *w* is the weight, *L* is the layer and *z* is the output.

The input for the CNN model is the final image features. The image data is fed into the neuron or a hidden layer that has weight, multiplying the input number with the neuron's weight which provides the output that is used as input for the next layer. In order for the inputs to be mapped into outputs that are needed for the network to function, an activation function is used. It is a non-linear activation function that

helps the network to learn complex data and provide accurate predictions. The structure of a CNN is illustrated in Figure 2.7.

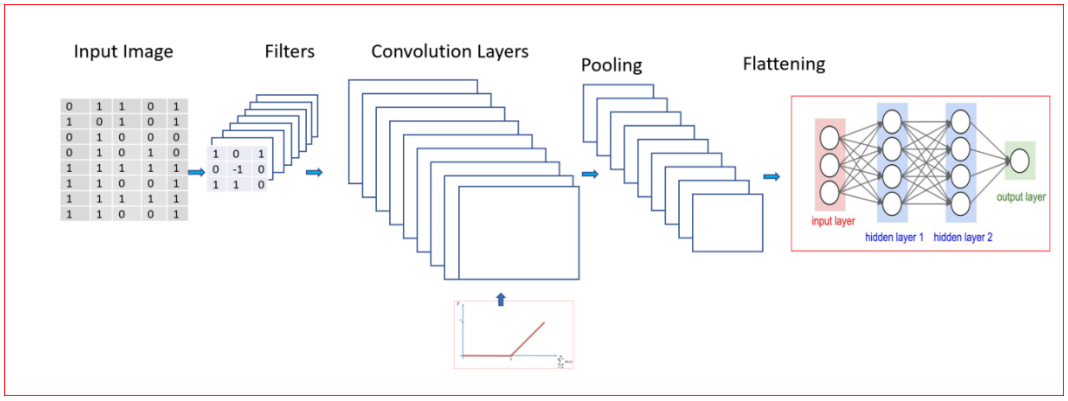


Figure 2.7: Representation of a full convolution neural network. Retrieved from (Khandewal,2018)

Some of the milestones in deep learning include; AlexNet, Zeiler network, DeepFace, the DeepID series, VGGFace and FaceNet. The DeepFace was centred on deep convolutional networks created by Facebook's research team that constituted of Parkhi et al. (2014). It was used to identify human faces in 2 million digital image faces. It used deep CNN feature extractors and used it to classify the images. The form of metric learning they used was that they trained the model to minimalize the distance between similar pairs of faces and maximise the distance between dissimilar pairs. It achieved an accuracy of 97% when tested on benchmark datasets.

DeepID was developed by Sun et al. (2014), it used Convolutional Neural Networks to extract features and Joint Bayesian or neural network for recognition. The last hidden layer instead of being used as the output it was used for features. Identities were classified simultaneously and not by binary classifier training. They gained an accuracy of 2.03% and 1.68% for Joint Bayesian and Neural network respectively. After benchmarks with Labelled Faces in the Wild database they achieved an accuracy of 94.32% and 96.05% for Neural Network and Joint Bayesian respectively. It was later improved in the following publications by training via

contrastive loss to improve identification and verification tasks. Figure 2.8 summarizes the evolution of face recognition from holistic to deep learning.

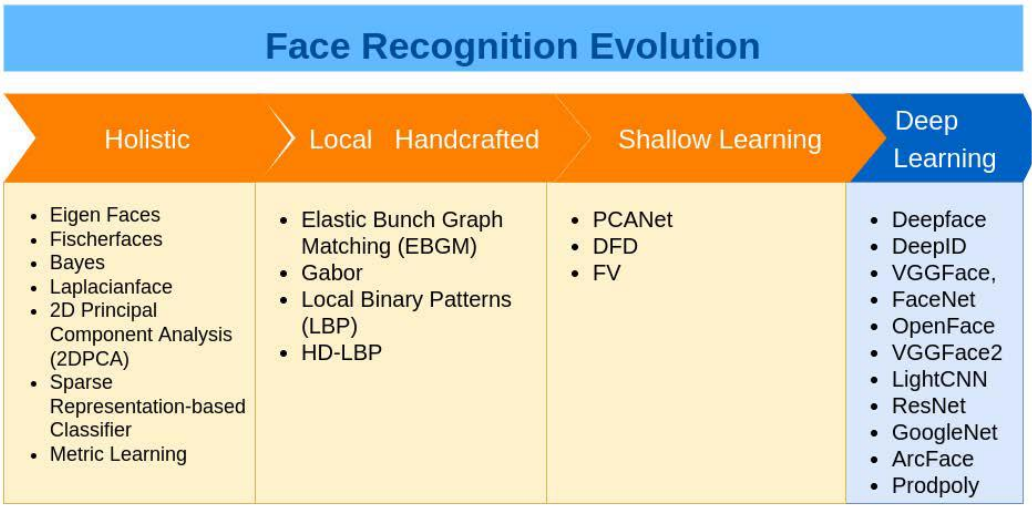


Figure 2.8: Evolution of face recognition methods retrieved from (Wanyonyi & Celik, 2022)

Other researchers have also used deep learning for face recognition in their work. Gao et al. (2019), proposed a real time, free of landmark estimation deep Siamese network algorithm that preserved identity during face synthesis. The algorithm used contrastive loss function, pose invariant features to perform face recognition, PCA for dimension reduction and LDA for classification. Face synthesis model was used to transform the non-frontal faces to virtually frontal faces. The Siamese network was used to remove the distortion of identities caused by face synthesis; therefore, identity was preserved during the process. The contrastive loss minimized the feature distance from the same object and enlarged it if they were from different object, hence, making the features more discriminative. Their network achieved superior performance compared to 2D deep learning-based algorithms.

Khan et al. (2019), proposed a framework for face detection and recognition that was based on convolutional neural network. It outperformed earlier techniques; it was protected, reliable and easily usable. It experienced challenges when it came to faces with increasing beard, glasses, tilted face and moustache. Over-fitting was also a

problem. Qi et al. (2019), proposed an algorithm that would solve the weakness of model generalization due to the necessity to use fixed-size images and the use of one network for extracting features. They used cascaded CNN which combined separable convolution and remaining structure in the network. Their algorithm achieved competitive accuracy to other techniques in real-time performance.

Kong et al. (2018), developed a deep CNN model based on CSGF $(2D)^2$PCA Net to solve data repetition, extensive computation period and rotation variations. They used circularly symmetrical gabor filter for rotation invariance, 2-D PCA for feature extraction. The model had two feature extraction stages and one non-linear output stage. The algorithm was robust to variations in occlusion, illumination, pose, noise and expression. Li et al. (2019), proposed a full graphical processing unit-based batch multi-task cascade convolutional network that would have a superior speed performance. Their algorithm performed better than multi-task cascade convolution network.

2.3 Partial Occlusion in Face Recognition

The facial recognition methods require the accessibility of a whole input face, lack of which may lead to wrong grouping (Satonkar et al., 2011). However, in the real-world environment human faces especially criminal's, are likely to be occluded. In the field of face recognition, face occlusion is one of the most challenging problems due to the lack of previous knowledge concerning the parts that are occluded. This becomes even more difficult to explore because of the parts can be any shape or size and anywhere in a face image (Zeng et al., 2021).

According to (Min et al., 2014), face occlusions can be caused by reasons that can either be undeliberate or intentional. For example, in security, scarves, sunglasses and caps are worn by criminals to ensure their faces are not recognized while committing illegal crimes (Min et al., 2014). They also added that, in order to achieve robustness in face recognition, it is crucial to control partial occlusion.

On the other hand, Towner and Slater (2007), classified face occlusion as systematic and temporary. Facial components such as scars, hair and ornamentations worn by people such as clothes, glasses, all fall under systematic occlusions. Whereas, temporary occlusions include but is not limited to temporarily obscuring a face with other objects, changes in environmental conditions such as shadows and lighting, changes in head pose cause self-occlusion and objects placed in front of the face temporarily also cause occlusions. In their study (Zhang et al., 2018), derived that interaction with the environment constantly because it is necessary from our daily lives results in causing self-occlusion more frequently than as compared to other temporary occlusions. Table 2.1 shows some of the categories of occlusions that exist.

Table 2.1: A categorization of occlusion challenges

Occlusion Scenario	**Examples**
Facial accessories	eyeglasses, sunglasses, scarves, mask, hat, hair
External occlusions	occluded by hands and random objects
Limited field of view	partial faces
Self-occlusions	non-frontal pose
Extreme illumination	part of face highlighted
Artificial Occlusions	random black rectangles
	random white rectangles
	random salt & pepper noise

Source: *Zeng et al. (2021)*

2.4 Problems with Partial Occlusions in Face Recognition

Face recognition under occlusions is difficult to solve. There are many reasons that make it so difficult. According to (Zhang et al., 2018), they include; occlusions vary considerably depending on the position where they occur in a face. Secondly, a face can have many types of occlusions. Thirdly, an occlusion's location may not be fixed on a face like hand covering; hence, it is difficult to predict their exact location.

Fourthly, occlusion duration varies in length depending on the type. For example, in a video a hand movement may last for a short period of time whereas an ornament like scarf or sunglasses may last longer. Finally, occlusion visual properties are unpredictable and small portions of the face are impacted which can be compensated with un-occluded parts of the face.

2.5 Face Recognition Approaches under Partial Occlusions

To mitigate the problem of partial occlusion in face recognition some approaches have been proposed for different occlusion scenarios.

2.5.1 Occlusion Scenarios

According to (Zeng et al., 2021) survey, there are many occlusion scenarios that tests are run on. These scenarios depend on the images on the gallery and probe or test set. These are; real occlusions which occur when the gallery images are free from occlusion but the test faces have occlusions such as scarves or sunglasses which are realistic in nature. Partially occluded faces are used when the gallery images are un-occluded whereas the test faces are partially occluded. When the images on the gallery are faces captured in the wild or from uncontrolled environment and the test aces use synthetic occlusions so to emulate realistic occlusion, this constitutes synthetic testing scenario.

Occluding rectangle testing scenario is used when the images on the gallery are un-occluded mugshots whereas the probe image faces have black and white rectangles occluding them. The last testing scenario that (Zeng et al., 2021) described was the occluding unrelated image scenario. It comprises of a gallery set with occlusion free mugshots whereas probe faces set have occlusions such as non-square image or an animal which are unrelated images.

2.5.2 Occlusion Robust Approaches under Face Recognition

Some approaches have been developed in an effort to counter the problems caused by occluded faces. Zeng et al. (2020), classified these approaches into three

categories. The first category is the occlusion robust feature extraction. This category focuses on the feature space that is not affected largely by face occlusions. For the cross-occlusion strategy learning-based and patch-based engineered features are utilized. The second category is the occlusion aware face recognition. Approaches in this category assume that visible parts are ready; therefore, during face recognition occluded parts are excluded. The third category is the occlusion recovery-based face recognition. Occlusion recovery is used as the cross-occlusion strategy in that the occlusion-free face is recovered from an occluded face.

2.5.2.1 Occlusion recovery-based face recognition approaches

These approaches work on the principle of recovering whole faces from the occluded faces hence, use face recognition algorithms directly. These methods use techniques such as reconstruction for face recognition or inpainting which considers the occluded face as a repair problem. Jia and Martinez (2008) proposed an approach to enable face recognition in both the training and test sets. By allowing occlusions in both the training and testing sets, they estimated the occluded test image as linear combination of the training samples of all classes. For reconstruction, non-occluded parts were used because the distinct face areas were weighted differently. In other words, they based their reconstruction on the visible data on the training and testing sets compared to previous works that focused on the testing sets. Their approach performed well on the AR dataset.

Iliadis et al. (2017), proposed a robust and low rank representation for fast face identification with occlusions. In their proposed framework, they wanted to solve the block occlusion problems by utilizing a robust representation that was based on two features because they wanted to model the contiguous errors.

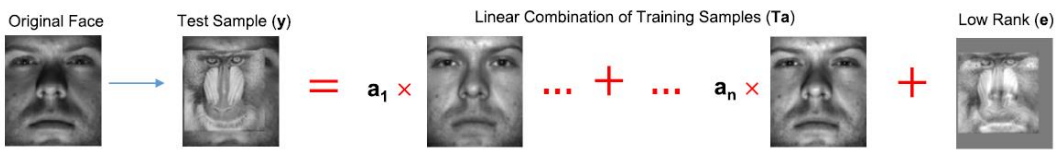


Figure 2.9: Representation of an occluded face sample as the linear combination of training sample with some intra-class variations plus the error. Retrieved from Iliadis et al. (2017)

The first feature used a loss function to fit the errors of Laplacian sparse error distribution, whereas the second described the error image or modelled it as low rank structural by obtaining the difference between a test face that is occluded and the training sample of the same identity that is un-occluded as shown in Figure 2.9. Their approach was efficient in computational time and identification rates.

Wang et al. (2017), proposed an occlusion detecting and image recovering algorithm. Occlusion detecting involved occlusion detection and elimination whereas the image recovery involved recovery of occluded parts and reservation of un-occluded parts. They used genuine and synthetically occluded face images. This approach produced global features that were good and beneficial to classification.

Vijayalakshmi (2017), proposed a way to recognize face with partial occlusions using In-painting. A partial differential equation method together with modified exemplar in-painting was used to remark the face region that was occluded. Despite the approach achieving recognition rate increases it had a limitation in that the image data used for the work was not representative of a real-world scenario.

According to (Wei et al., 2014) facial decorations such as scarf, veil objects, sunglasses and reduced image quality blurring can cause occlusion. As a result, it affects face recognition in that the discriminative facial features are misleading and the distance between the two face images of the same subject is enlarged in feature space. This results in intra-class variations becoming higher than inter-class variations. Additionally, occlusion of facial landmarks leads to the existence of registration errors hence degrading the recognition rate.

Wei et al. (2014), proposed a dynamic image to class warping (DICW) framework that used local matching-based approaches to solve the problem of face occlusions. The local matching-based approach mined facial features from local areas of the

face. The affected and unaffected parts of the face could be analysed in isolation. The matching errors were minimised through subspace, partial distance and multi-task sparse representation learning strategies (Wei et al., 2014). Matching of the faces was completed by outlining a distance measurement between sequences and using the distance as the base for classification. (Wei et al., 2014) used Euclidean and cosine for pixel intensity and LBP feature local distance measurements respectively. They used 2,400 samples for each occluded versus un-occluded, un-occluded versus occluded and occluded versus occluded scenarios. Their framework had an 96.7 % correct identification rate on the AR database that had over 4,000 colour images of 126 subject's faces, 97.3 % on the AR dataset without alignment, a 0.8740 area under curve on the LFW database under unsupervised learning setting.

2.5.2.2 Occlusion robust feature extraction approaches

These approaches use methods such as handcrafted features such as LBP, SIFT and HOG descriptors. One of the advantages of such methods is the easiness that comes with extraction of features from raw images. Additionally, their discriminative and tolerance to large variability and also being computationally efficient since they lie low in the feature space constitutes to more advantages (Zeng et al., 2021). On the other hand, they have limitation in that for face recognition, integration of the decision from local patches is required and also for frontal faces, alignment based on eye coordinates contributing to precise registration. In other words, the need for face images to be aligned well so that features can be extracted hinders its application in real life (Zeng et al., 2021).

Learning based features methods such use learning-based approaches to extract features have been proposed. These methods include linear subspace, sparse representation classification and non-linear deep learning methods. These methods have succeeded because of the characteristics such as smooth surface and regular texture that face images have in common compared to regular images. For discrimination among features, subspace learning preserves variation in faces (Zeng et al., 2021). This has been applied by Eigenfaces (Turk & Pentland, 1991), using principal component analysis (PCA).

Another method that has been used is Fisherface using LDA (Zeng et al., 2021). Fisher face was first introduced by (Etemad & Chellapa, 1997). It works by learning a class specific transformation matrix. Its performance depends heavily on the input data. It is a supervised dimension reduction algorithm. It allows reconstruction of an image but a nice reconstruction is impossible because, the features had already been discriminated before. It is also an enhanced Eigen face method that for dimensionality reduction, it uses Fisher's Linear Discriminant Analysis (LDA) whereby, the LDA works better in discrimination than PCA in that the ration between a class scatter to within a class scatter is maximized. It's good when the face images have illumination and facial expression variations (Ismail & Sabri, 2009). The Eigenface and Fisherface methods have a disadvantage in that there's a need for the eye location to be aligned properly. This is not the case in real world data.

For the occlusion possibility to be accounted for, statistical learning methods are used. Methods such as self-organizing maps projections that takes into account that probability of occlusions occurring is different dependent on the occlusion. Another approach is the sparse representation classifier. With the goal that a representation that accounts for occlusion and corruption is generated, training samples and sparse errors are combined linearly.

Wu and Ding (2018), proposed a low-rank regression with generalized gradient direction to suit occluded face recognition. Dictionary learning sparse representation was used in combination with low rank representation on the error term leading to a low rank optimization problem as shown in Figure 2.10.

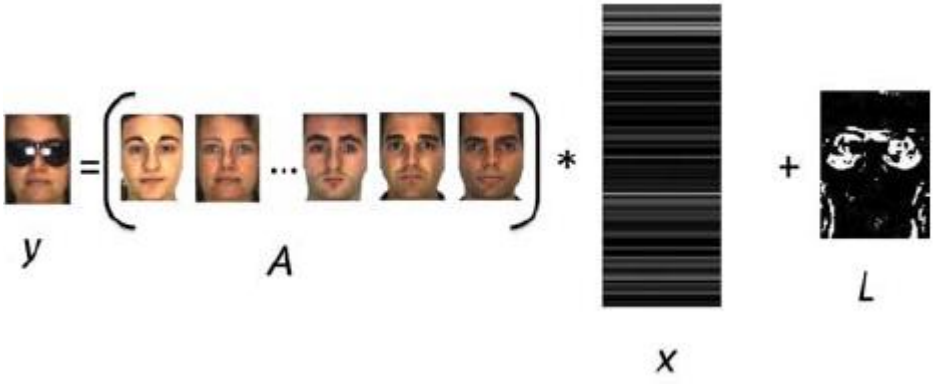


Figure 2.10: Representation of the occluded image *y*, by a linear combination of all training data in the dictionary *A* and added by a residual image *x* standing for occlusion whereby *L* is the residue as retrieved from Wu and Ding (2018)

Their approach had robustness to any size, type and kind of occlusion like shadows, objects on the face and achieved good performance compared to the state-of-art frameworks at the time.

Yang et al. (2017), proposed a joint and collaborative representation with local adaptive convolution feature. With their aim being able to achieve robust face recognition under occlusions, they used CNNs to learn convolution features extracted from local regions that were discriminative to the face identity. Their approach exploited the uniqueness and commonness of the different local regions as shown in Figure 2.11. Their experiments showed that varying local regions have varying discrimination, furthermore, some local regions never contribute to and sometimes they may even mislead the face recognition.

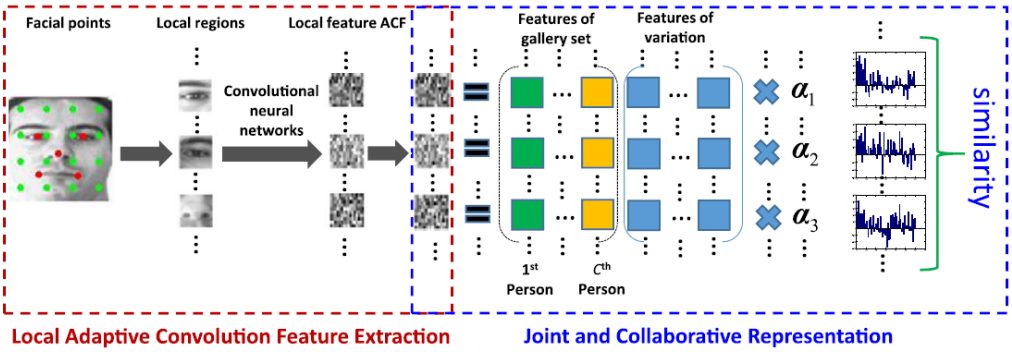


Figure 2.11: Flowchart of the joint and collaborative representation with local adaptive feature model as retrieved from Yang et al. (2017)

Another approach in this category is deep learning. If a massive training dataset having enough occluded faces is provided for a deep network, then occlusion robust face recognition is achieved (Zhou et al., 2015). One of the milestones in deep learning is the FaceNet model. FaceNet was developed by google researchers (Schroff et al., 2015). It was a data driven system in that they used a large dataset of labelled faces which enabled them attain pose, illuminations and other variations and it attained advanced results with benchmark datasets. The limitation was that it was data driven and this is not always the case in practical scenarios.

Cen and Wang (2019), proposed a deep dictionary representation-based classification (DDRC) that was to improve robustness in face recognition with occluded faces. They used an already trained CNN for feature extraction, whereby, they performed a nonlinear mapping from the image space to the deep feature space.

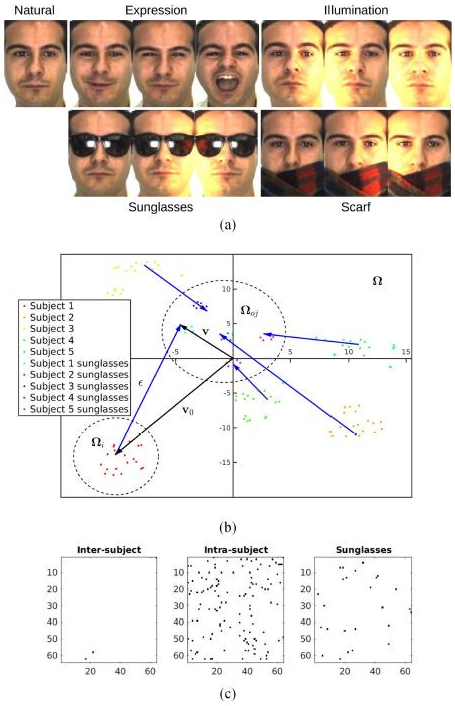


Figure 2.12: Deep feature vectors represented visually from 5 subjects from the AR dataset. Retrieved from Cen and Wang (2019)

Figure 2.12 shows the representation of the deep feature vectors. Whereby (a) represents example images, (b) represents 2-D visualization of the deep feature vectors and (c) Visualization of the deep feature components activated commonly for the natural faces associated with different subjects (inter-subject), the non-occluded faces associated with the same subject (intra-subject) and the sunglasses-occluded faces associated with different subjects (sunglasses).Whereby V_0 represents deep feature vectors of the occluded face and V represents deep feature vectors of the un-occluded face Ω_i represents the subspace associated with non-occluded face images of the i^{th} subject Ω_{oj} represents the subspace associated with face images occluded with j^{th} occlusion pattern (Cen & Wang, 2019).

By defining the deep feature vector of a subject having a small error, as a linear combination of the column vectors of the matrix defining all the deep feature vectors of the same subject in the training samples, a dictionary representation was achieved. The subject with deep feature vector was figured out through the identification of the

best approximation of the deep features within the subject's subspace. Therefore, a gallery was formed by concatenating the matrices of the deep features (Cen & Wang, 2019). Additionally, a regularization constraint was enacted when the gallery's size was greater than the feature measurement.

Auxiliary dictionaries for the proposed DDRC were formed through concatenation of auxiliary dictionaries of some identified types of occlusions (Cen & Wang, 2019). This was done because a test sample's occlusion pattern was unknown. To obtain a unique solution a regularization restriction and the squared Euclidian norm minimization were used to obtain the estimated coding coefficients which were in turn used to recover the deep feature vector of the occluded face. Finally, classification was done by comparing the similarity of the occluded face of the subspace to the non-occluded face of the subject. The overfitting problem was solved by the use of PCA for dimensionality reduction.

The DDRC had a 94.7%, 85.7%, 99.3% and 98.7 % on the AR database with faces with occlusion in the first session, occlusion in the second session, scarf session one and scarf session two respectively. A 98.6%, 94.5%, 88.2% and 54.5% accuracy rates on the FERET database with block occlusion ratio of 9.7%, 19.1%, 28.9% and 39.1% respectively. A 91%, 76.1%, 62.4% and 46.8% recognition rate on the CelebA database dataset with block occlusion ratio of 0%, 4.98%, 10% and 14.7% respectively. This approach had a limitation in that it assumed the test faces occlusion patterns were included in the auxiliary dictionary, hence limiting its usage.

Mao et al. (2019), developed a framework that utilized the gradient and the shape cues in a deep learning model to detect and verify occluded faces. Since the head-shoulder location resembles an omega which is also similar to a Gaussian curve, they suggested that a local minimal energy could be reached by adding suitable energy terms. Head scanning was done using defined curved lines which had a movement rule and corresponding energy values on the left, right and upper scanning lines respectively. Their detection algorithm had three phases: "position initialization, minimizing a potential energy based iterative process and energy constraint

conditions". They employed ellipse fitting to solve the problem of not getting the accurate lower jaw position.

In verifying whether a human face was covered or not, (Mao et al., 2019) employed a sparse classifier with deep CNN features. Due to the difficulty in collecting enough samples, they built a deep model that depended on less samples and a dictionary learning framework to learn more effective features. The SRC model they developed had an additional similarity constraint to seek correlations between similar descriptors through a shared dictionary space. This in turn made the dictionary more discriminative for the classification task on which it was based on. By assuming that the dictionary and sparse code values were constant, the sparse model was optimized. In their experiments, they used a stationary digital video recorder (DVR) which simulated bank system monitoring. The experiments showed that the head detection algorithm performed at 98.89% accuracy rate whereas, the designed occlusion verification scheme achieved a 97.25% accuracy rate.

2.5.2.3 Occlusion aware face recognition approaches

One of the approaches in this category is the occlusion detection-based face recognition. To tackle the occlusion problem, these methods first perform occlusion detection and later a representation is obtained from the non-occluded parts (Zeng et al., 2021). The other approach is the partial face recognition, based on the assumption of the availability of a partial face and uses it for face recognition and the occlusion detection stage is not considered. In other words, this approach focuses on the face recognition stage and avoids the face occlusion detection stage. Partial faces can often be found in real world data such as in mobile devices or surveillance cameras (Liao et al., 2013).

For occlusion detection, items such as sunglasses and scarves are used as a representation of occlusion because of their frequent appearance in the real world. For the visible parts selection, it is done through the assumption that previous knowledge of occlusion is known hence, skipping the face occlusion detection phase. (Song et al., 2019), proposed a pairwise differential Siamese network (PDSN) that

was to capture the relationship between the occluded facial block and corrupted feature elements, thereafter, establish a mask generator as shown in Figure 2.13.

The PDSN developed by Song et al. (2019), consisted of a trunk CNN and a mask generator branch forming a Siamese architecture. The mask generator module was expected to output a mask whose element was a value between 0 and 1. Upon multiplication of the mask value with the input contaminated feature it could diminish its corrupted elements. This was learnt through minimizing a combination of two losses; classification and pairwise loss. A fixed mask was extracted from every trained mask generator and a dictionary was built because the trained PDSN could not be directly used to output the feature discarding mask (FDM) of a probe face. Through the combination of relevant dictionary items, the FDM of the face with arbitrary partial occlusions was derived. It showed significant improvement in the performance on face recognition on both the real and synthesized face datasets.

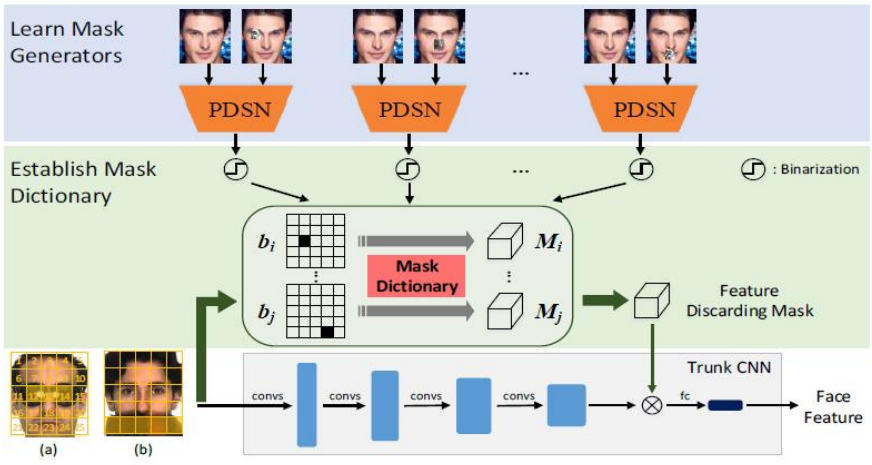


Figure 2.13: The overview of the PDSN framework where b_i b_j the non-overlapping face blocks, M_i and M_j binarized feature discarding mask. Retrieved from (Song et al., 2019)

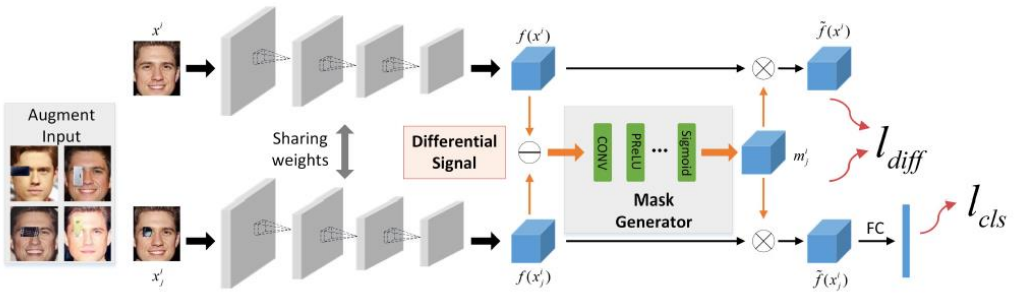


Figure 2.14: The pairwise differential siamese network. Retrieved from (Song et al., 2019)

The pairwise differential Siamese network is illustrated in Figure 2.14 whereby; x^i is the non-occluded face image, x^i_j is the occluded face image, $f(x^i)$ is the clean feature, $f(x^i_j)$ is the remaining part of the feature after masking, $f(x^i)$ is the corrupted feature, $f(x^i_j)$ is the top conv feature of an occluded face after masking, $l_{cls\ classification}$ is the loss, l_{diff} is the differential signal and pairwise loss.

In their experiments the PDSN had an identification accuracy of 95.84%, 97.29%, 97.36%, 98.26%, 97.98% and 97.92% on a discarding threshold of 0, 0.05, 0.15, 0.25, 0.35 and 0.45 on the AR dataset with sunglasses and scarf occlusions. A 54.80% and 56.34% identification accuracy rate on the with and without differential supervision information on the Facescrub probe set. A 99.20% face verification rate on the LFW benchmark. A 74.40% face identification accuracy on the MegaFace challenge and a 98.19% and 98.33% on the AR dataset with sunglasses and scarfs natural occlusions respectively.

For partial face recognition, (Liao et al., 2013), proposed an alignment-free approach in partial face recognition as shown Figure 2.15. Their proposed method did not require any alignment of the face's focal points. For the representation of a partial face with variable length, they employed multi-key point descriptors. A dictionary was constructed from the descriptors from a gallery that was large; hence, the descriptors of the probe image were represented sparsely and inferred the identity of the probe image. It had a limitation in practical application because; the number of faces required by SRC to cover all variations is quite high.

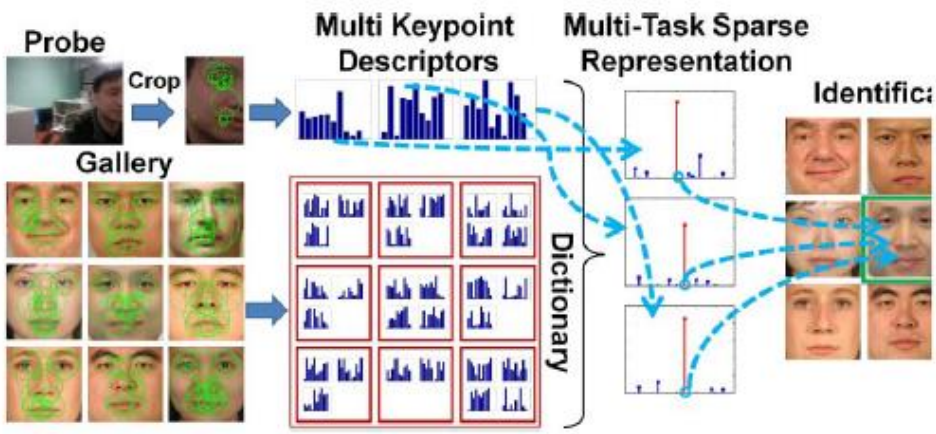


Figure 2.15: An illustration of the alignment free approach as retrieved from (Liao et al., 2013).

2.6 Summary

The above literature review clearly showed that the state-of-the-art face recognition systems or frameworks still experience problems with occlusion while testing on different datasets as also shown in Table 2.2.

Table 2.2: Summary of occlusion approaches' strengths and weaknesses

Approach	Techniques	Strengths	Weaknesses
Occlusion recovery	Reconstruction (Jia & Martinez, 2008)	Computationally efficient	Some of the image data used for these works are not representative of a real-world scenario
	In-painting (Vijayalakshmi, 2017)	Increased identification rates	
Occlusion robust feature	Handcrafted features such as LBP, SIFT and HOG descriptors (Zeng	Tolerant to large variations	Requires alignment based on eye coordinates

extraction	et al., 2021)		
		Computationally efficient	For deep learning, the amount of data required is too large in a real-world scenario
	Learning based features such as linear subspace, sparse representation classification and non-linear deep learning methods (Yang et al., 2017), (Wu & Ding, 2018), (Cen & Wang, 2019)	Performance improvements	
Occlusion aware	First perform occlusion detection and later a representation is obtained from the non-occluded parts (Song et al., 2019) Partial face recognition-based on the assumption of the availability of a partial face and uses it for face recognition and the occlusion detection stage is not considered (Liao et al., 2013).	Significant improvement in recognition rates	Data required to cover all variations of occlusions is quite high

A framework based on occlusion aware face recognition approaches was proposed. It was loosely based on the Song et al. (2019) assumption that human visual system ignores occlusion and solely focuses on the non-occluded sections for recognition. Another motivation was, because occlusions in real world data are so many one cannot be able to train a model with all the occlusions for identification. Hence, models that discard occlusions can be used for practical applications.

2.7 Conceptual Model

A conceptual model was derived based on the proposed detection and exclusion of occluded regions approach.

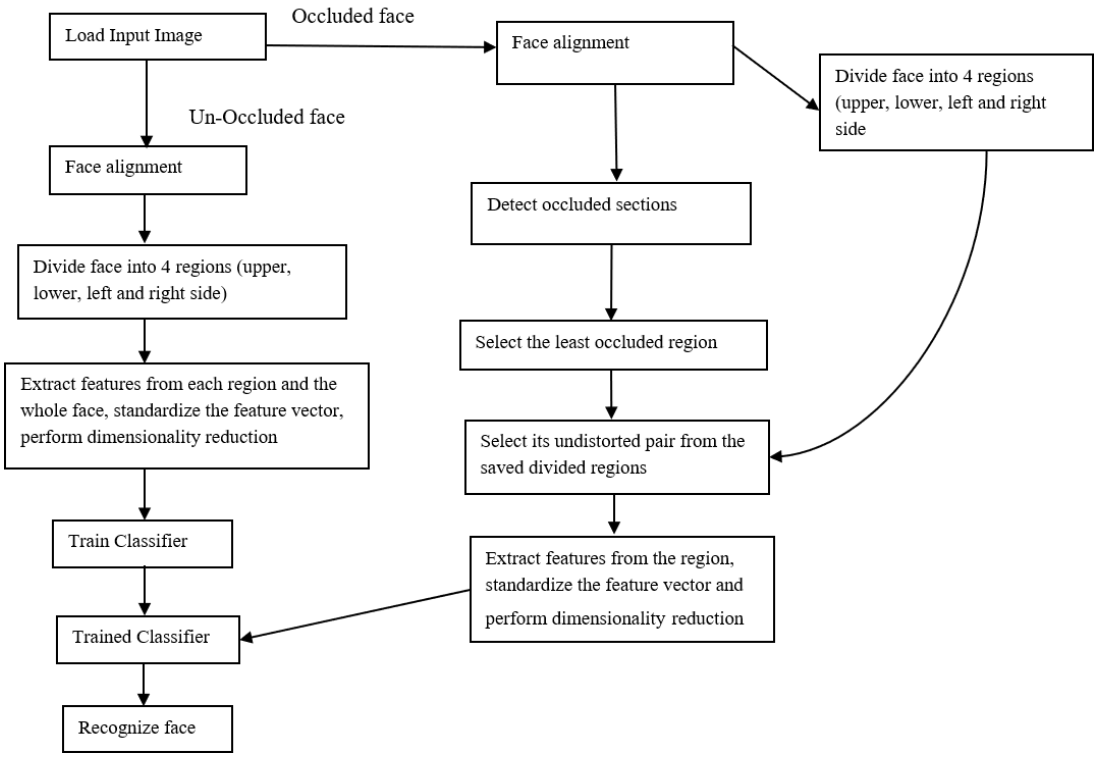


Figure 2.16: Conceptual model of the proposed approach

Therefore, in this research, the proposed approach would be designed so as to advance the performance of face recognition in partial occlusion scenarios. Figure

2.16 illustrates the conceptual model for the proposed approach. For training, face images with no occlusions would be used. They would be divided into sections; the upper, lower, left and right sections. These sections would be used in training a classifier. For testing, the faces with occlusions such as sunglasses and masks would be used. The occluded parts would be discarded and the non-occluded parts would be used for face recognition. This would be useful in criminal identification using face recognition because criminals often hide part of their faces to avoid being recognized.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discusses the research methodology. An experimental research design was adopted. The proposed approach adopted, data collection, experimental set-up and ethical considerations are discussed.

3.2 The Proposed Approach

This research was based on the assumption that a partially occluded face can be recognized because the human visual system focuses on the non-occluded parts of the face (Song et al., 2019). In the proposed approach, a trained model or classifier would be derived from features of whole/un-occluded face images. Thereafter, the trained model would be used to identify or classify occluded face images. Therefore, the aim of this research was to develop a face recognition approach that would recognize faces from the non-occluded sections of an occluded face image as shown in Figure 3.1.

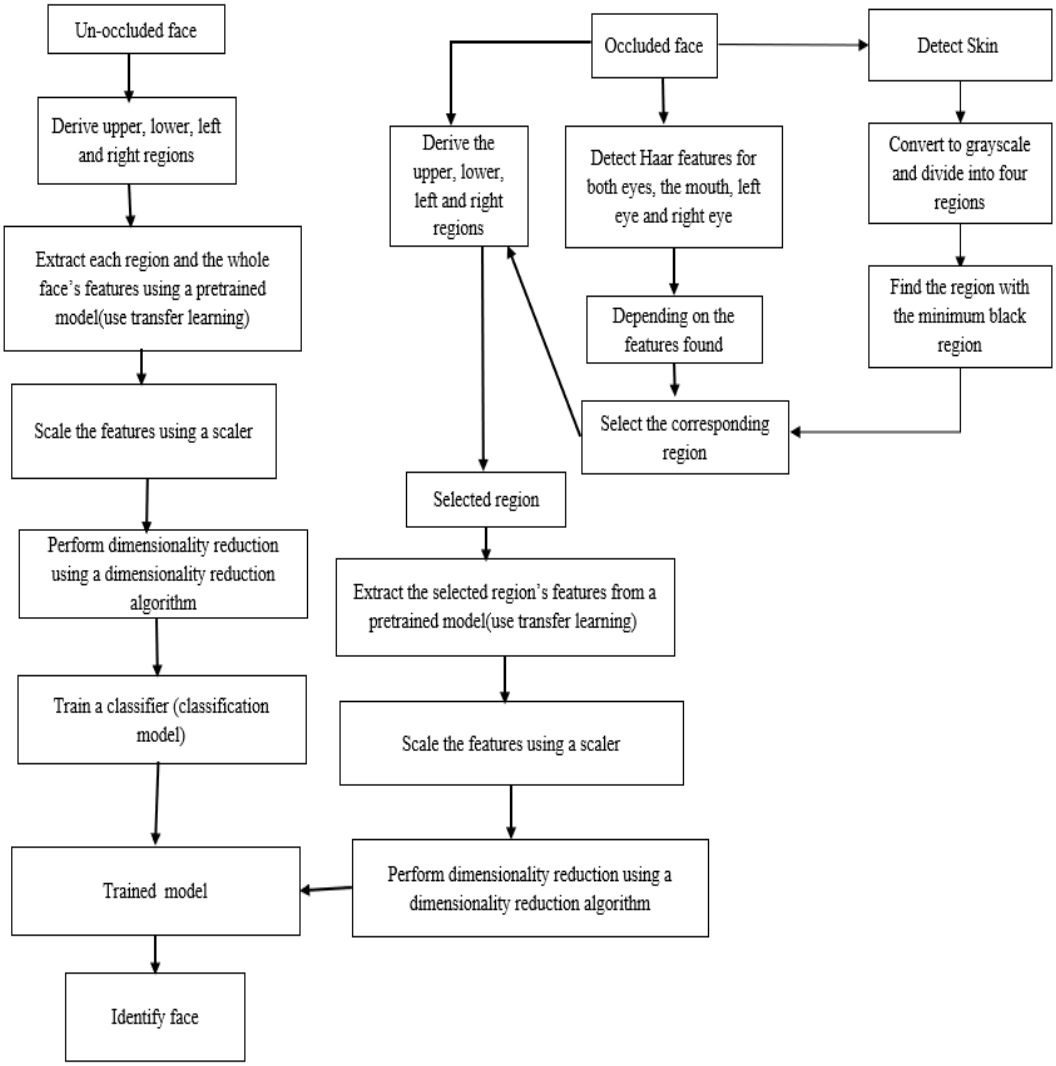


Figure 3.1: Representation of the approach process flow

To develop a classification model using a whole/un-occluded face; An aligned and cropped whole/un-occluded face image was denoted as a Cartesian plane; the height of the image as the y-axis and the width as the x-axis as illustrated in Figure 3.2.

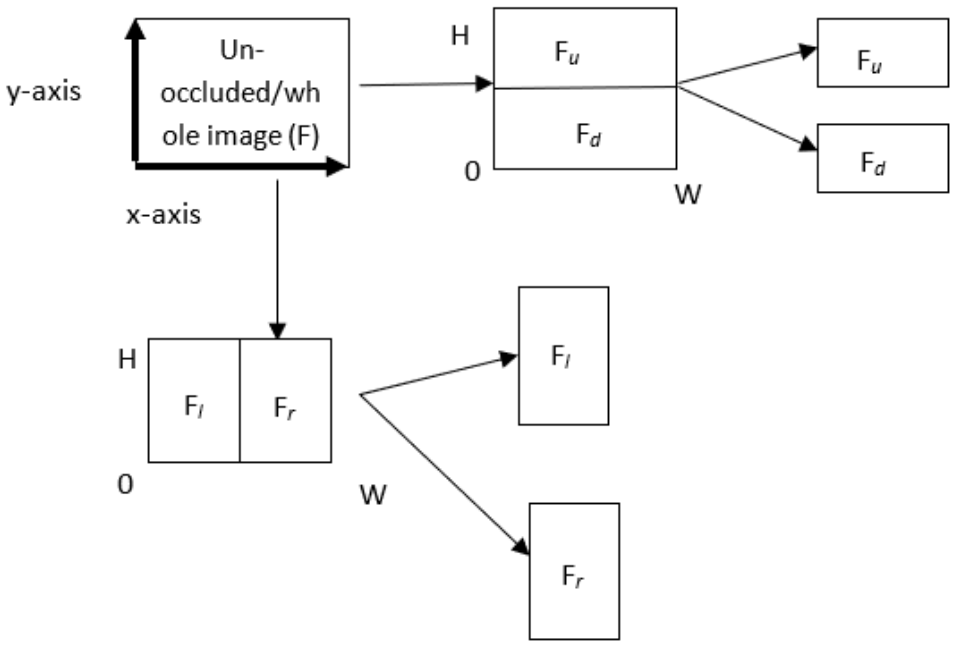


Figure 3.2: Representation of the face as a Cartesian plane

Using these axes; dividing the image equally along the y-axis (height)/horizontally produced the upper section F_u and the lower section F_d of the face whereas dividing it equally along the x-axis (width) /vertically produced the left section F_l and right sections right section F_r of the face as shown in Figure 3.3. The height of the F is denoted as H and the width of F was denoted as W. Given the origin of F as (0,0):

$$F_u = \left(0, \frac{H}{2}\right), (W, H) \quad (1)$$

$$F_d = (0,0), \left(W, \frac{H}{2}\right) \quad (2)$$

$$F_l = (0,0), \left(\frac{W}{2}, H\right) \quad (3)$$

$$F_r = \left(\frac{W}{2}, 0\right), (W, H) \quad (4)$$

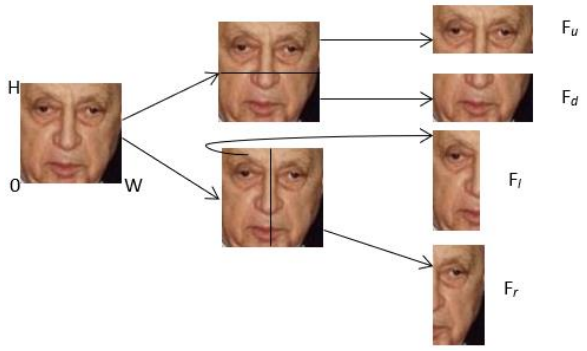


Figure 3.3: An example image from the Pubfig dataset used to demonstrate equation (1) to (4)

Given a pre-trained model as M, a feature vector V is derived by passing the original and all other augmented face images through it. Representing each face image as F_m ;

$$V = M \rightarrow F_m \qquad (5)$$

After feature extraction, the features are normalized or scaled and dimensionally reduced. Given V_s as the scaled feature vectors and V_d as the dimensionally reduced feature vector, a classifier C is trained hence generating a trained model T_m.

$$V_s = \text{scaler}(V) \qquad (6)$$

$$V_d = \text{dim\_reduction}(V_s) \qquad (7)$$

$$T_m = C \rightarrow V_d \qquad (8)$$

Where; scaler represents a scaling or normalisation algorithm and dim_reduction represents a dimensionality reduction algorithm.

To identify or recognize an occluded face images from the trained classifier/model derived from equation (8); the occluded face O, is divided into four sections as in equation (1) to (4) above to generate; O_u for the upper section, O_d for the lower section, O_r for the right section and O_l for the left section of the face as shown in Figure 3.4. Additionally, an illustration of this is given in Figure 3.5.

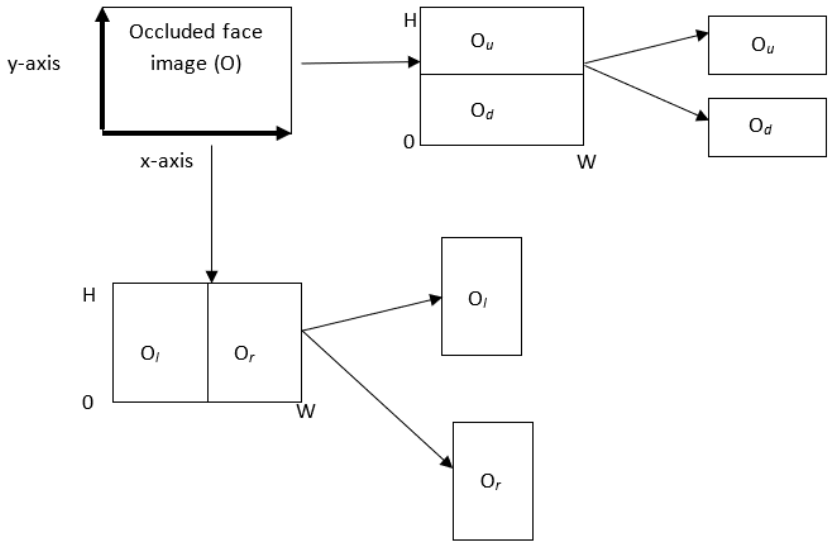


Figure 3.4: Deriving O_u,O_d,O_l,O_r from the occluded face

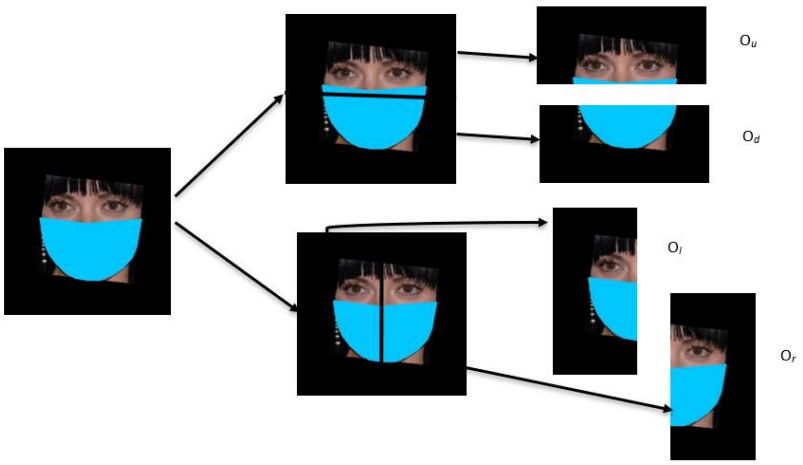


Figure 3.5: Deriving Ou, Od, Ol, Or from a synthetically occluded image

The O was used to detect the non-occluded section of the face, so that it can be used for the identification of the face. The strategies that were used to detect the non-occluded section were; skin detection and haar cascade classifiers. Skin detection can be defined as a way of finding the skin-coloured regions and pixels in an image whereby the skin colour is the primary identifier of the skin (Kolkur et al., 2017). Skin colour can be segmented using saturation and value (HSV) and luma component, blue component and red component (YCbCr) colour spaces. In this research a combination of the watershed algorithm and the two-colour spaces was used (Jean, 2018).

A watershed algorithm is used to perform object segmentation on grayscale images. Viewing a grayscale image as a topographic surface; high can be denoted as a peak whereas low intensity as a valley. To perform separation on the images, each valley is filled out with water of different colours (Ray, 2020). As the water rises slowly, water from different valleys will start to merge. Barriers are built in the locations where water merges to avoid the merging. The work of filling water and building barriers continues until all the peaks are under water. The segmentation result is derived from the barriers created (OpenCv, N/A).

After skin detection and segmentation as illustrated in Figure 3.6, the image was converted to grayscale. The grayscale image was divided into the four regions as discussed above and each region was converted to an array and its black or the 0 values elements were summed up. The least occluded region was selected by identifying the minimum value of the four summed up elements' values from the four regions.

Figure 3.6: A skin segmented face image

The least black region = The least occluded region

On the other hand, a haar cascade classifier is an object detection algorithm (Mittal, 2020) that was proposed by (Viola & Jones, 2001). In this research, pre-trained haar cascade classifiers for the eyes, right eye, left eye and mouth were used. Some of the objects or regions that were detected were both eyes, to detect visibility of the upper section of the face, the mouth, to detect visibility of the lower section of the face, the right eye, to detect the visibility of the right section of the face and the left eye to detect the visibility of the left section of the face.

Depending on the region that was non-occluded U_r, its corresponding section from the divided face sections was selected as the input to generate a feature vector from the pre-trained model. For example, if both eyes are detected; O_u is selected or if lower section has a lot of skin than other regions O_d is selected as illustrated in Figure 3.7.

$$U_r = \min (O_u, O_d, O_r, O_l) \qquad (9)$$

Figure 3.7: The least occluded region selected from Figure 3.5

After deriving the feature vector from equation (5), the test sample feature vector was scaled as in equation (6) and the result projected to the dimension space of equation (7) to generate the final feature vector V_{ur} that would be used to predict the identity of the face I_f as the output.

$$I_f = T_m \rightarrow V_{ur} \qquad (10)$$

3.2.1 The proposed detection and exclusion of occluded face regions approach algorithm

The algorithm that was derived from the proposed approach was;

1. **Input:** Un-occluded aligned and cropped face images Fs
2. **For** each face F
3. Divide the face image into four sections;
   Given **H** as the height of the face and **W** as its width and origin as (0,0);

$$F_u = \left(0, \frac{H}{2}\right), (W, H)$$

$$F_d = (0,0), \left(W, \frac{H}{2}\right)$$

$$F_l = (0,0), \left(\frac{W}{2}, H\right)$$

$$F_r = \left(\frac{W}{2}, 0\right), (W, H)$$

4. **For** each variation and original faces from (3) as F_m
5. pass it through a pre-trained deep learning model, M to extract feature vectors, V
6. Scale the feature vectors, V_s
7. Perform dimensionality reduction on V_s to produce V_d
8. **end for**
9. **end for**
10. Train the classifier, C using V_d
11. **Output** a model T_m, $T_m = C \rightarrow V_d$
12. Face recognition
    a. **Input:** An occluded face image O
    b. Divide the image into four regions; upper, lower, left and right sections
       Given H as the height of the face and W as its width and origin as (0,0);

$$O_u = \left(0, \frac{H}{2}\right), (W, H)$$

$$O_d = (0,0), \left(W, \frac{H}{2}\right)$$

$$O_l = (0,0), (\frac{W}{2}, H)$$

$$O_r = \left(\frac{W}{2}, 0\right), (W, H)$$

c. Use the whole occluded image from (a) to detect the non-occluded section using haar cascade classifiers or skin segmentation
d. **if** the features are found in a region
e. **return** region for example: O_u as U_r
f. **end if**
g. Select the corresponding region of (e) from (b)
h. **if** for example $U_r = O_u$ in equation (e)
i. O_u from equation (b) is selected
j. **end if**
k. Use the selected region/section from (i) to retrieve a feature vector from the pre-trained model, V_u
l. Scale the V_u and perform dimensionality reduction by projecting the V_u it to the equation (6) dimension space to produce the V_{ur}
m. **Output:** identity of the face I_f from trained classifier T_m from (8) and the features V_{ur} from (l)

$$I_f = T_m \rightarrow V_{ur}$$

3.3 Data Collection

A lot of data (face images) are required to evaluate performance of face recognition approaches. Collecting such data is time consuming and labour intensive. Secondly, since this research thesis was focused on improving existing algorithms as reviewed in the literature, choosing to work with the datasets used therein was important for performance comparison. There were several publicly available datasets for evaluating face recognition tasks but fewer in evaluating face recognition in partial occlusion scenarios.

The publicly available datasets that were collected were the Webface-OCC (Huang et al., 2021), the labelled faces in the wild (LFW) (Huang et al., 2007), Public Figures Face Dataset (Pubfig) (Kumar et al., 2009), the FaceScrub dataset (Ng & Winkler, 2014) and the extended Yale Face Database B (Yale B) (Georghiades et al., 2001) that was later modified to be used in this thesis and a custom dataset. The summary of data collected is shown in Figure 3.1. The Webface-OCC dataset contains images

with simulated or synthesized occlusions. It has 804,704 face images of 10,575 subjects (Huang et al., 2021).

The LFW dataset is set up under unconstrained environment and designed for unconstrained face recognition tasks. The dataset contains more than 13,000 face images captured under various environmental conditions collected from the web. This dataset has very few occluded face images. Images from the Pubfig dataset have large variations in parameters, scene, lighting, pose and imaging conditions since they were taken from uncontrolled environment and the subjects were not cooperative.

Table 3.1: Summary of Data Collected

Dataset No	Dataset Name	Images	Subjects	Source
1.	Webface-OCC	804,704	10,575	Authors
2.	Labelled Faces in the Wild (LFW)	13,000	5,000	Kaggle
3.	Pubfig	11,790	150	Kaggle
4.	FaceScrub	45,760	526	Github
5.	Extended Yale B	16,128	28	Authors

3.4 Experimental Setup

This section describes the tools and algorithms used in running these experiments. These include the system specifications, development tools, face alignment, data augmentation, feature extraction, classifier and occlusion detection algorithms used. For the Webface-OCC, Pubfig, FaceScrub and Yale B datasets, an experiment was

carried out using the skin detection and haar cascades for occlusion detection and retrieval of the non-occluded section. While using the LFW dataset, the validation protocol (Huang et al., 2007) was followed.

3.4.1 System Specifications

The experiments were implemented on a virtual machine with Intel Xeon 5118 2.3GHz CPU, 8 GB RAM, and an Ubuntu operating system.

3.4.2 Development Tools

The development tools that were used for these experiments were Python (python org, N/A), as the programming language. Python comes with a lot of inbuilt libraries like scikit-learn for machine learning and open-cv for face recognition. The Pycharm community edition 2021 (Jetbrains, N/A), an integrated development environment (IDE) was used. This Pycharm IDE is great for pure python development and it is also free and open-source.

3.4.3 Data Pre-processing

Pre-processing of data involved cropping, trimming and labelling images that had not been cropped nor labelled before. The images were later saved with their specific label, and then saved into labelled folders referred to as classes depending on their label. These labelled folders were finally saved into two folders those are: the training and testing folders. The final two folders were saved into one main folder labelled after its dataset name; for example, Webface-OCC dataset.

3.4.4 Selecting the Data Sample

Data splitting is the process of dividing a dataset into training and testing sets. This process is important because it helps to prevent overfitting of the model. Overfitting of a model happens when a model fits exactly against its training data. This leads to a model having a low train error and a high-test error. In other words, the model fails

to perform accurately on new or unseen data as expected (IBM Cloud Education, 2021).

A standard train-test split method could not be used in this research because it was approached differently and there was an imbalance in the datasets. The train data would only contain whole faces, whereas the test data would contain partially occluded faces. Therefore, the train test ratio was not of major significance in this research.

The Webface-OCC dataset has both the occluded and un-occluded faces. 20 classes were selected randomly for this research. For the 20 classes, at most 20 whole face images were used per class for training, 10 occluded face images for validation and another 10 occluded face images for testing the model. The Pubfig, FaceScrub and Yale B datasets did not have occluded faces. This meant that some of the faces therein had to be synthetically occluded for this research. 20 classes from the Pubfig and FaceScrub datasets were selected for this research.

On the other hand, all the classes (28 classes) in the Yale B dataset were used. For the training data, at most 20 face images were set aside for it, at most 10 face images for testing the approach. In all the datasets, the classes included images of both male and female subjects of various races and ages. The LFW dataset was used to validate the approach generally using its validation protocol. For benchmark comparison, they recommend using 10-fold cross validation with splits they have randomly generated for evaluation and averaging the results. The results such as accuracy and receiver operating curve (ROC) are used.

3.4.5 Face Alignment

Face alignment is a process that ensures that the facial landmarks of a face image are aligned spatially. This process has been used to increase the face recognition rate because the geometric variations of a face are effectively reduced (Wei et al., 2020). The steps in face alignment include: face detection using landmark localization,

thereafter, using the landmarks such as eye regions to rotate, translate and scale the face.

The dlib library (King, 2021) was used to detect the facial landmarks. The dlib library is a c++ toolkit that contains several machine learning algorithms. For this research the dlib's image processing tools for frontal face detection and shape prediction were used. The frontal face detector function is configured to find human faces that look more or less towards the camera. After face detection, the FaceAligner class's align method from imutils library (Rosebrock, 2021) that is publicly available was used to align the face. These libraries were downloaded from the internet and used in the experiments. All the faces selected for the experiment in 3.4.3 were aligned.

3.4.6 Masking Faces from the Pubfig, FaceScrub and Yale B Datasets

The Pubfig, FaceScrub and Yale B datasets did not contain a lot of partially occluded faces. Therefore, there was a need for the datasets to be partially occluded synthetically. An algorithm by (Mein, 2020), was used to overlay masks on the selected test face images. To overlay a mask on the face image; A face image is first resized into 500 pixels, detecting its face and facial landmarks, thereafter using these landmarks to overlay a mask on the face image. This has been illustrated in Figure 3.8 and Figure 3.9.

Figure 3.8: Face that is not masked retrieved from the Facescrub dataset (Ng & Winkler, 2014)

Figure 3.9: A synthetically masked face

3.4.7 Data Augmentation

Due to the small size of data per class that would be used for training a classifier, at most 20 face images, there was a need to augment the data to create more data in a class. Data augmentation is the process of artificially creating new training images from existing training images. This can help reduce overfitting and improve generalization (Dvornik et al., 2021). A data augmentation algorithm (Jung, 2021), was used for this experiment. The algorithm is used to augment images for machine learning experiments. Data augmentation functions with corruptions such as noise, Gaussian blur, zoom blur were used as shown in Figure 3.10**.**

Additionally, other functions such as image sharpening, image multi hue saturation, adding canny edges, embossing and blend alpha were also used. To find out how many images are required to learn a class; 5, 10 and 15 augmentations were added to each image. In other words, 500%, 1000% and 1500% of images were added.

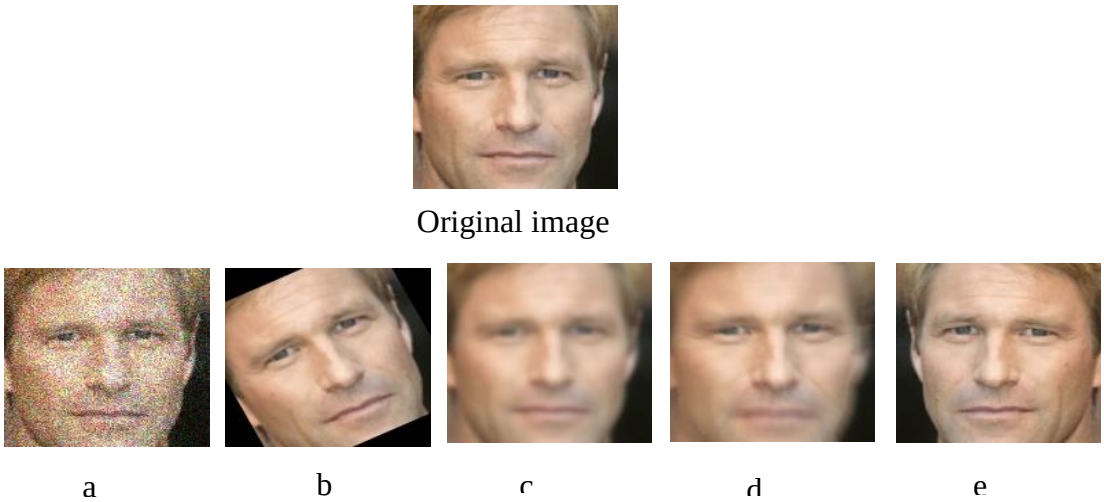


Figure 3.10: Original face image from FaceScrub dataset (Ng & Winkler, 2014) with its augmentations

An example of the image augmentations that are done are shown in Figure 3.10 whereby; image (a) represents the shot noise corruption, (b) rotation, (c) gaussian blur, (d) zoom blur and (e) flipped augmentations.

3.4.8 Feature Extraction

This research adopted the concept of transfer learning for feature extraction. Transfer learning is important in developing classification models because it is rare to train a model from scratch due to the amount of data required (Chilamkurthy, NA). Therefore, pre-trained deep convolutional networks were used for feature extraction. For feature extraction, a pre-trained model's weights are loaded. Once an image is passed through the model it returns a representation of the image as a feature vector.

In this research, three pretrained models were used. First, the VGG (vgg11) (Simonyan & Zisserman, 2015) pre-trained model was selected to extract features. A variation of the model was also used by (Cen & Wang, 2019) in their experiments.

The VGG pre-trained models are publicly available and they have demonstrated human level accuracy on the LFW dataset. The vgg11 model has 11 weighted layers, whereby 8 are convolutional layers and 3 are fully connected layers. It requires an RGB image of size 224 * 224 and the output is a 1000 feature vector from its third fully connected layer as shown in Figure 3.11.

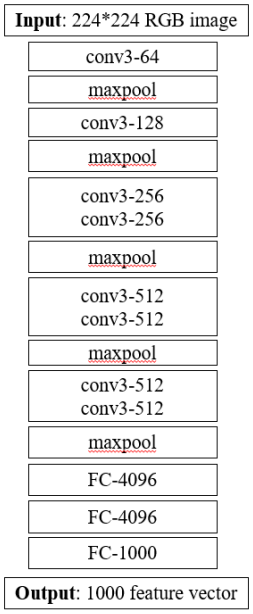


Figure 3.11: The vgg 11 input, layers and output

For comparison purposes on which feature extractor model would work best, we secondly used another torch vision pretrained model ResNet (resnet18) (He et al, 2015) and thirdly, the Inception Resnet (V1) model from (Esler, 2021) pre-trained on the VGGFace2 dataset. The resnet 18 model, illustrated in Figure 3.12 has 18 layers and requires an input image of size 224 * 224 and the output is 1000 feature vector.

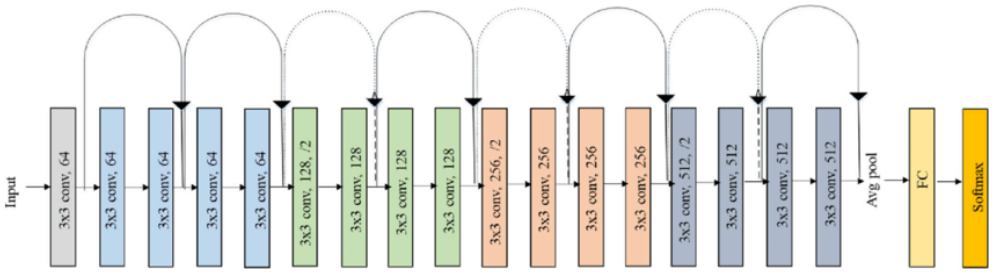


Figure 3.12: Resnet-18 architecture retrieved from (Razman et al., 2019)

On the other hand, the Inception Resnet (V1) model is a hybrid model of inception and Resnet model. It incorporates 3 different stem modules and reduction modules. Its architecture is shown in Figure 3.13. It generally requires a 299 * 299 image but for this experiment, a square face image of dimensions 160 * 160 was used as input to the pre-trained model whereby a 512-feature vector was retrieved.

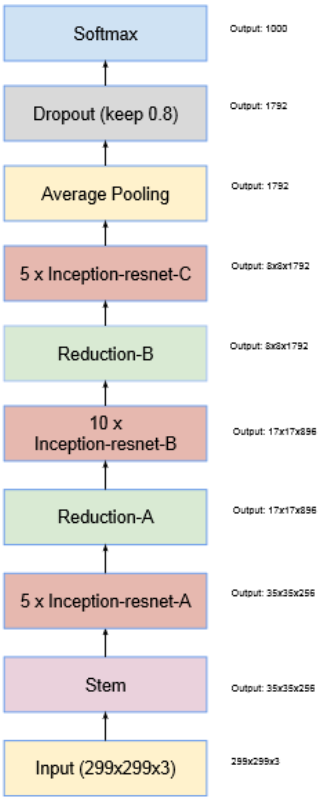


Figure 3.13: Schema for Inception-Resnet-v1 and Inception-ResNet-v2 networks Retrieved from (Szegedy et al., 2016)

3.4.9 Feature Normalization and Dimensionality Reduction

The feature vectors that were extracted from the pretrained models were scaled using the min max scaler from (Pedregosa et al., 2011). The min max scaler method was fit with the features and the derived scaler fit was used to transform the train and test

data into normalized data. Thereafter, dimensionality reduction was performed using linear discriminant analysis (LDA) (Pedregosa et al., 2011), to extract the most relevant features for training a classifier. The scaled features were projected to the LDA space and depending on the classes used, the features retrieved equalled to the total number of classes minus 1. Dimensionality reduction is important because it helps remove random noise that is independent, that is, it's not correlated with the input and the label. It also removes unwanted degrees of freedom in that the input can change without the label changing (Wang & Carreira-Perpinan, 2014).

3.4.10 Classifier Training

A supervised learning approach was used to train the models because we had labelled datasets. We adopted both the linear and non-linear classifiers and the process followed the stages outlined in Figure 3.14.

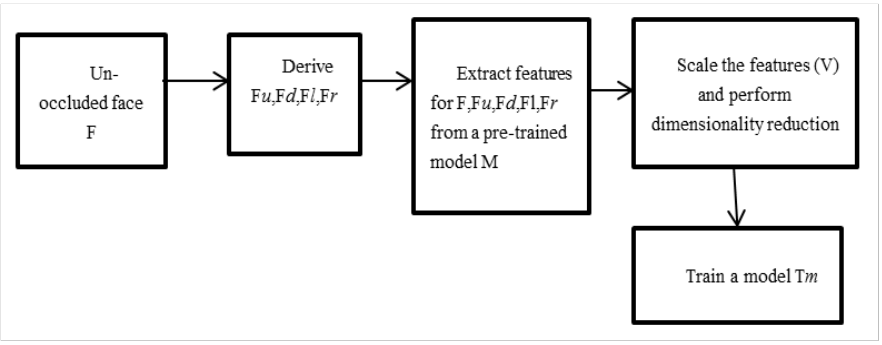


Figure 3.14: Training phase flowchart

Three classifiers were chosen for training. The Linear discriminant analysis (LDA) and multi-layer perceptron (MLP) classifiers from the scikit-learn library (Pedregosa et al., 2011) and a custom multi-layer perceptron (Custom MLP) built from scratch for performance comparison with the scikit-learn classifiers.

A linear discriminant analysis classifier finds a linear combination of features that separate classes of objects. It is based on the assumption that each class's Gaussians share the same covariance matrix. In this research it was used as both a

dimensionality reduction algorithm and a classifier. It uses the Bayes' rule for predictions. The LDA classifier can be represented mathematically by;

$$P(y = k|x) = \frac{P(x|y = k)P(y = k)}{P(x)} = \frac{P(x|y = k)P(y = k)}{\sum_l P(x|y = l) \cdot P(y = l)}$$

Whereby; P(X|y = *k*) is the data for each class *k*

On the other hand, a multi-layer perceptron is an algorithm that relies on the underlying neural network and can learn a non-linear function approximate for classification provided a set of features and targets. These two models work well with small datasets and are easily accessible from the scikit library.

3.4.11 Trained Classifier/ Model Evaluation

The proposed approach was based on the (Song et al., 2019) assumption on the availability of a partial face in a partially occluded face image. Hence it employed the occlusion aware approach. Therefore, given a partially occluded face, the non-occluded section had to be detected and selected so that it could be used for recognition. Two strategies were used to detect occlusion and discard it; skin detection and the use of haar cascade classifiers to detect both eyes, mouth, left eye and right eye. The detection and exclusion of the occluded face regions followed the process shown in Figure 3.15.

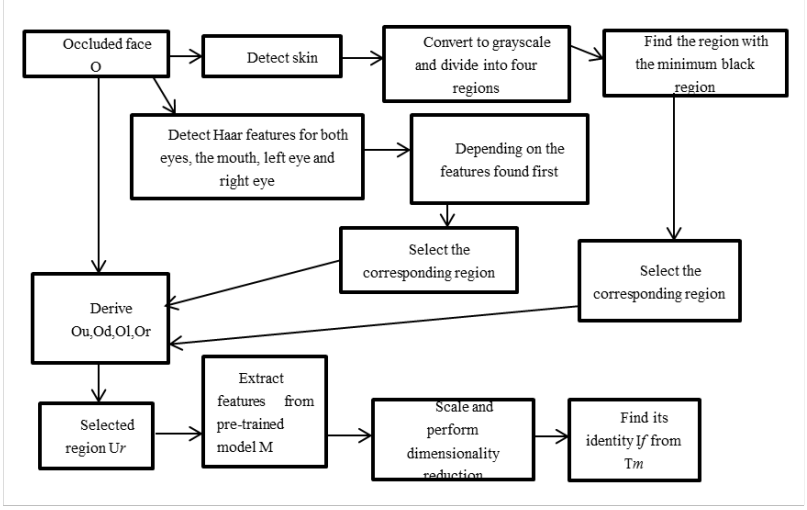


Figure 3.15: Testing Phase flowchart

To evaluate the classifier, we used some evaluation metrics. Evaluation metrics can be defined as the quantifiers of the performance of a predictive algorithm. Since this research was done to solve a classification problem, classification metrics such as accuracy, precision, recall and F1 score were used.

3.4.11.1 Accuracy

This metric measures the percentage of correct classifications given test data.

Whereby; Accuracy = total number of correct predictions/total number of predictions

3.4.11.2 Precision

This metric measures the ratio between the true positives and all positives. It is used to measure all relevant data points and is also referred to as specificity of the model.

Precision = TP/(TP+FP)

Where TP = true positive, FP = false positive

3.4.11.3 Recall

This metric measures the ability of a classification model to correctly identify true positives and ability to identify relevant data. It is also referred to as the true positive rate or sensitivity.

$$Recall = TP/(TP+FN)$$

Where TP = true positive, FP = false positive, FN = false negative

3.4.11.4 F 1 score

It is used to show how precise and robust the classification model is and is based on the precision and recall metrics. The best value of an F 1 score is 1 and the worst value is 0.

3.5 Ethical Considerations

Face recognition software use human face data for development. Such data is sensitive and may infringe the face image owners' privacy. It is therefore crucial to observe ethical considerations before using such data. The datasets used in these experiments have been licensed for academic use only. This research abides by the regulations in those licenses.

3.6 Conclusion

In this chapter, the methodology adopted for this research has been discussed. The discussion included, the research design whereby this research adopted an experimental design. The data collection process, whereby, secondary data was used. The experimental setup; from data selection, feature extraction to training and evaluating models. Finally, the ethical considerations were highlighted.

CHAPTER FOUR

EXPERIMENTS' RESULTS ANALYSIS AND DISCUSSION

4.1 Introduction

In this chapter, results from the experiments conducted using the methodology discussed in the previous chapter are discussed. The detection and exclusion of occluded regions approach was an occlusion aware approach and based on the assumption by (Song et al., 2019) that is always a partial face in an occluded face image that can be identified by the human visual system. Therefore, we had to retrieve the partial section of the face that is least occluded or non-occluded region and use it for face recognition. To achieve this, we employed two methods; skin detection and haar cascades. Three pretrained models; resnet 18, vgg 11 and inception resnet V1 were used as feature extractors. Additionally, three classifiers; linear discriminant analysis (LDA), multilayer perceptron (MLP) and a custom multilayer perceptron (Custom MLP) were trained and evaluated in these experiments. The trained classifiers were evaluated using the accuracy, precision, recall and F1 score classification metrics.

4.2 Experiments' Results and Analysis

The experiments were split into two based on the least or non-occluded region retrieval method. Thereafter, for each experiment, the three feature extraction models were also used.

4.2.1 Non-Occluded Region Retrieval using Skin Detection

A non-occluded region from the occluded face image was retrieved using the skin detection algorithm as described in the previous chapter.

4.2.1.1 Using resnet18 as a feature extractor

For these experiments the resnet18 pretrained model from torchvision (Torch Contributors, 2022) was used.

Table 4.1: Recognition rates using resnet18 and skin detection

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	LDA	51	56	51	51
	MLP	48	49	48	46
	CUSTOM MLP	49	53	49	47
Pubfig	LDA	40	41	40	38
	MLP	39	40	39	37
	CUSTOM MLP	44	50	44	44
FaceScrub	LDA	55	57	55	55
	MLP	53	56	53	53
	CUSTOM MLP	53	56	53	52
Yale B	LDA	24	35	24	24
	MLP	26	34	26	26
	CUSTOM MLP	19	28	19	18

The performance of the models as shown in Table 4.1 using features extracted from the resnet18 pretrained model were low on all the datasets. The features that were extracted from the resnet18 the model had a negative transfer learning effect. Therefore, the resnet18 model could not be used as feature extractors for this approach.

4.2.1.2 Using vgg11 as feature extractors

For these experiments the vgg11 pretrained model from torchvision (Torch Contributors, 2022) was used.

Table 4.2: Recognition rates using vgg11 and skin detection

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	LDA	51	60	51	50
	MLP	58	60	58	56
	CUSTOM MLP	51	53	51	50
Pubfig	LDA	42	49	42	42
	MLP	43	47	43	43
	CUSTOM MLP	45	56	45	46
FaceScrub	LDA	48	50	48	47
	MLP	47	49	47	46
	CUSTOM MLP	50	53	50	49
Yale B	LDA	27	37	27	27
	MLP	26	34	26	27
	CUSTOM MLP	26	37	26	27

The performance in Table 4.2 using features extracted from the vgg11 pretrained model were too low compared to state-of-art models. There was negative transfer learning using vgg11 as a feature extractor model for this approach. Therefore, the vgg11 model could not be used as feature extractors for this approach.

4.2.1.3 Using Inception Resnet (V1) as a feature extractor

For these experiments the inception resnet V1 pretrained model (Esler, 2021) was used.

Table 4.3: Recognition rates using Inception Resnet (V1) and skin detection

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	LDA	87	90	87	87
	MLP	85	89	85	85
	CUSTOM MLP	82	88	82	83
Pubfig	LDA	97	97	97	97
	MLP	95	96	95	95
	CUSTOM MLP	96	97	96	96
FaceScrub	LDA	92	93	92	92
	MLP	91	92	91	92
	CUSTOM MLP	91	91	91	91
Yale B	LDA	84	88	84	84
	MLP	80	84	80	80
	CUSTOM MLP	82	85	82	82

Table 4.3 shows results that were obtained using the Inception Resnet V1 pretrained model and skin detection as the occlusion detection method. From the results above it's very clear that the model performed well on the testing set. The varying scores of precision, recall and F1 score in the test set shows that the model performed lower compared to the results in Table 4-6. The results were lower because some of the synthetically occluded images in the test sets had been occluded by objects coloured in the same colour space as human skin. Therefore, the detection of the non-occluded section or human skin failed for some images leading to wrong classification.

4.2.2 Non-Occluded Region Retrieval using Haar Cascade Classifiers

For these experiments haar cascade classifiers were used to detect the non-occluded region/section of the occluded face as described in the previous chapter.

4.2.2.1 Using resnet18 as a feature extractor

For these experiments the resnet18 pretrained model from torchvision (Torch Contributors, 2022) was used as a feature extractor.

Table 4.4: Recognition rates using resnet18 and haar cascade classifiers

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	LDA	25	31	25	23
	MLP	21	25	21	18
	CUSTOM MLP	18	22	18	16
Pubfig	LDA	39	41	38	39
	MLP	39	39	39	37
	CUSTOM MLP	41	48	41	41
FaceScrub	LDA	49	50	49	48
	MLP	47	50	47	47
	CUSTOM MLP	48	51	48	47
Yale B	LDA	30	40	30	29
	MLP	27	38	27	27
	CUSTOM MLP	23	32	23	23

The performance of the models as shown in Table 4.4 are low for face recognition problems. There was a negative learning effect using the resnet18 model features. Therefore, the resnet18 pretrained model could not be used as feature extractors for this approach.

4.2.2.2 Using vgg11 as a feature extractor

The vgg11 pretrained model from torchvision (Torch Contributors, 2022) was used as a feature extractor in these experiments.

Table 4.5: Recognition rates using vgg11 and haar cascade classifiers

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	LDA	24	25	24	22
	MLP	31	38	31	30
	CUSTOM MLP	25	38	25	25
Pubfig	LDA	40	48	40	40
	MLP	41	45	41	40
	CUSTOM MLP	44	52	44	44
FaceScrub	LDA	44	44	44	42
	MLP	43	45	43	42
	CUSTOM MLP	46	44	46	44
Yale B	LDA	40	51	40	40
	MLP	37	47	37	36
	CUSTOM MLP	35	40	35	34

The performance of the models as shown in Table 4.5 using features extracted from the vgg11 pretrained model was low compared to state of art models. The features had a negative learning effect. Therefore, the vgg11 model could not be used as feature extractor for this approach.

4.2.2.3 Using Inception Resnet (V1) as a feature extractor

The inception resnet V1 pretrained model (Esler, 2021) was used as a feature extractor for these experiments.

Table 4.6: Recognition rates using Inception Resnet (V1) and haar cascade classifiers

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	LDA	92	93	92	92
	MLP	90	93	90	90
	CUSTOM MLP	90	93	90	90
Pubfig	LDA	96	97	96	96
	MLP	94	94	94	94
	CUSTOM MLP	96	96	96	96
FaceScrub	LDA	92	94	92	92
	MLP	90	91	90	90
	CUSTOM MLP	90	91	90	90
Yale B	LDA	96	97	96	96
	MLP	94	95	94	94
	CUSTOM MLP	95	96	95	96

Using the haar cascade classifiers showed improved results as shown in Table 4.6 compared to the results in Table 4-3 on the same datasets and feature extractor. The haar cascade classifiers depend on the objects being detected and not the human skin. Therefore, the colour of the objects used for occlusion did not affect the performance of the cascade classifier. On the other hand, the results are a bit lower in other datasets compared to the skin detection because haar cascade classifiers misclassify objects from time to time. The values of accuracy, precision, recall and F1 score show that the classifiers/models trained well.

4.2.3 Performance on the LFW Dataset

To evaluate the performance of the proposed approach on the LFW dataset, we used their validation protocol. We therefore, did not use partially occluded face images nor detect occluded face regions. The experiment was done for a general comparison of the proposed approach to that of the state-of-the-art models. We used the pairs dataset set for verification which gave a 92% accuracy rate. The performance was lower compared to the state-of-the-art models.

4.2.4 Recognition Rates under Different Block Occlusion Coverage Area

The effect of occlusion on face recognition rates was looked into in this research. For this experiment a black box was randomly placed on the test images to cover 30 to 80 percent of the faces. These test images were used to test the performance of the approach under different occlusion percentages. The results are shown in Figure 4.1.

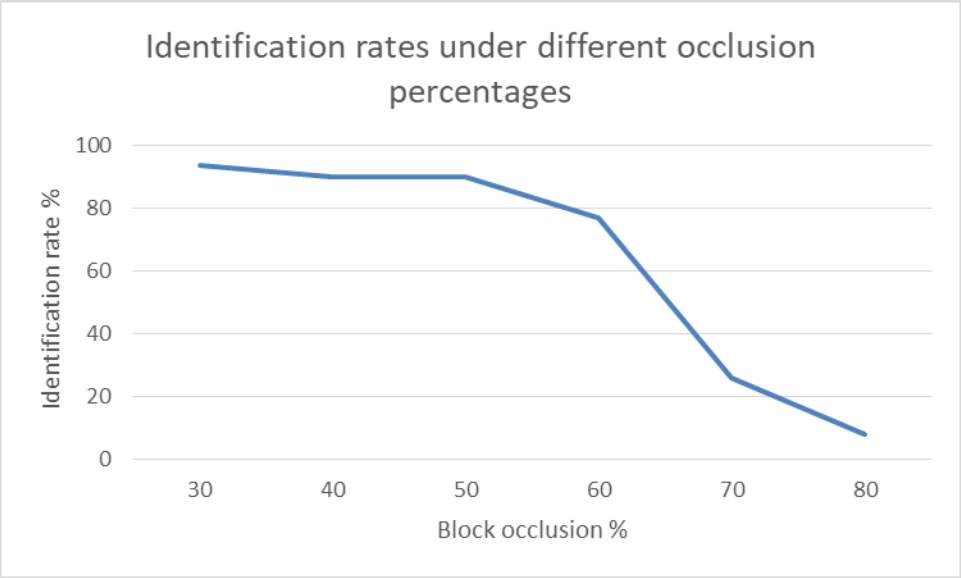


Figure 4.1: Recognition rates with different block occlusion percentages on the FaceScrub dataset

Figure 4.2: A face image retrieved from the FaceScrub dataset with 80% block occlusion

The chart in Figure 4.1 above shows the relationship between block occlusion and identification/recognition rate. An example of an occluded face image with 80% block occlusion is shown in Figure 4.2. There is a steady decrease in recognition

rates as the block occlusion area increases. As the block occlusion area on the face increases, the quality of usable features that can be used for recognition also decreases significantly. From the figure above the recognition rate decreased from 94% at 30% occlusion area to 8% at 80% occlusion coverage area. Therefore, occlusion affects face recognition rate significantly.

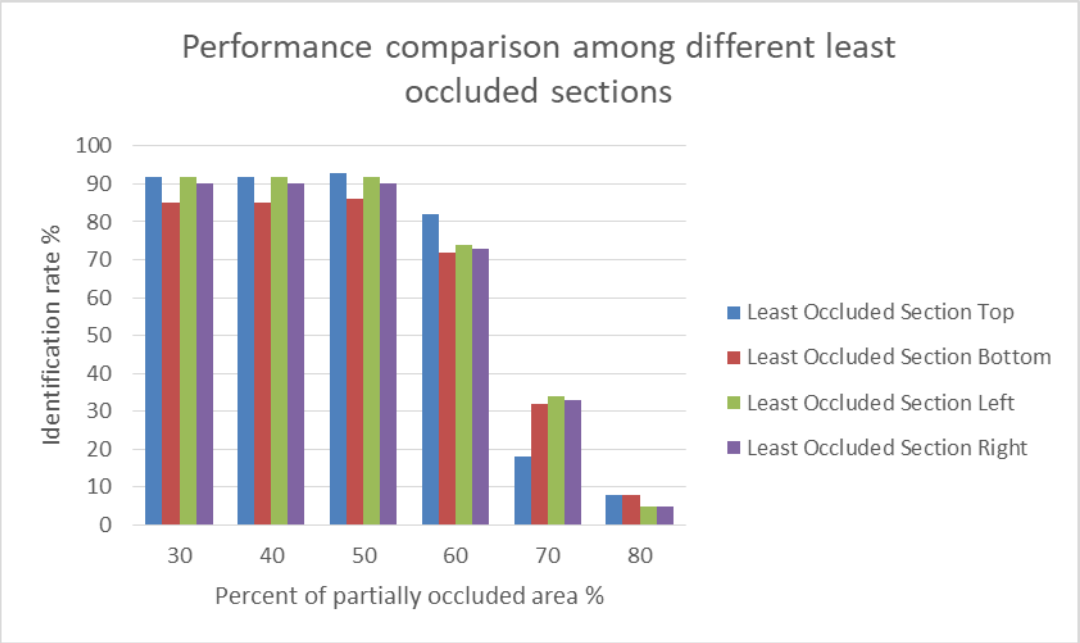


Figure 4.3: Performance comparison among different least occluded face sections

A performance comparison among different least occluded regions was done as shown in Figure 4.3. The performance comparison was among, the top section of the face that consists of some part of the nose, both eyes and forehead. The bottom section, that contains part of the nose and cheeks, mouth and chin. The left section of the face that has the left eye and left cheek and the right section of the face that has the right eye and right cheek. Figure 4.3 shows that when the top section of the face is least occluded it performs better than all other sections.

The top sections of the face contains both eyes and a study by Keil (2009) found out that the brain used the eyes as the principal source of information for face recognition followed by the mouth and nose. Additionally, eyes are less prone to

noise compared to other parts of the face. Therefore, the upper section gave better results because, the CNNs are developed to extract features similar to the way human visual cortex does. Additionally, this could also explain why the left and right sections of the face performed better as they contain an eye compared to the bottom section of the face.

4.3 Discussion

This research was aimed at developing a face recognition approach robust to partial occlusions. For this to done the existing face recognition approaches robust to partial occlusions had to be investigated and their strengths and weaknesses identified as discussed in chapter two. Based on the findings in chapter two an approach from existing approaches was developed as defined in the methodology section in chapter three. Thereafter, the performance of the face recognition approach to partial occlusions developed was evaluated as was done in this chapter by using measures of accuracy, precision, recall and F1 score.

The developed approach was based on the assumption by (Song et al., 2019) that human visual system ignores occlusion and solely focuses on the non-occluded sections for recognition. It was based on an occlusion aware approach; therefore, the occluded parts were discarded during recognition. The detection and exclusion of occluded regions adopted two methods; use of haar cascade classifiers (Alexey, 2015) and use of skin detection (Jean, 2018).

Transfer learning was adopted for feature extraction using pretrained deep neural networks as done by (Cen & Wang, 2019). Three deep CNNs pretrained models; resnet18 (He et al, 2015), vgg11(Simonyan & Zisserman, 2015) and Inception Resnet V1(Esler, 2021) were used. Three classifiers were used for training; the linear discriminant analysis (LDA), the multi-layer perceptron (MLP) and a custom multi-layer perceptron (Custom MLP). The approach was evaluated on the Webface-OCC, Pubfig, FaceScrub, Yale B and LFW datasets. The developed approach was evaluated using accuracy, precision, recall and F1 score classification metrics.

4.3.1 Summary of Key Findings

The results obtained suggests that using haar cascade classifiers as the occlusion detection method, use of a pretrained model, Inception Resnet (V1) as a feature extractor and used linear discriminant analysis (LDA) as a classifier yields the highest performance in terms of accuracy, precision, recall and F1 score among all datasets as evidenced in Table 4.6. There was significant improvement in performance using Inception Resnet (V1) as the feature extractor model. This could be attributed to the architecture of the Inception Resnet (V1) model uses multiple kernels for different sizes in one layer for effective recognition of variable sized features. Secondly, it could be due to the dataset used in training the model; the Inception Resnet (V1) used in this experiment was pretrained on the VGGFace2 dataset whereas the vgg11 and resnet18 are pretrained on the ImageNet dataset.

Therefore, the final approach was built based on; using Inception Resnet(V1) as a feature extractor, haar cascade classifiers as the occlusion detection method and use of Linear Discriminant Analysis (LDA) as the classifier/model. The detection and exclusion of occluded face regions approach is shown in Figure 4.3.

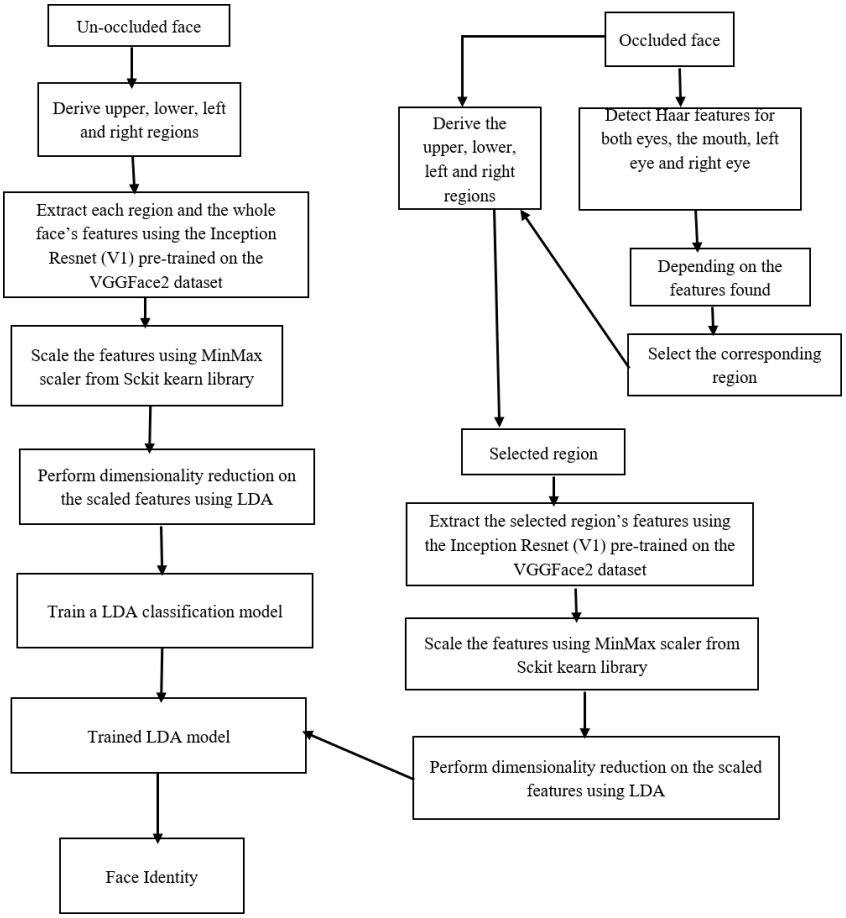


Figure 4.4: The detection and exclusion of occluded face regions approach

The developed detection and exclusion of occluded face regions approach was evaluated using the accuracy, precision, recall and f1 score classification metrics. The results obtained are shown in Table 4.7.

Table 4.7: Performance of the detection and exclusion of occluded face regions approach

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Webface-OCC	92	93	92	92
Pubfig	96	97	96	96
FaceScrub	92	94	92	92
Yale B	96	97	96	96
LFW	92	-	-	-

For all the experiments that were run on the different datasets, the accuracy, precision, recall and F1 score had high values, close to 100% and the values were almost similar. This meant that the approach performed well in the classification task. These results, therefore, support the assumption by (Song et al., 2019) that a face can be recognised by excluding the face's occluded regions and focusing only on the non-occluded region.

The effect of occlusion on face identification/recognition rates was also observed as shown in Figure 4.1. The rate of face recognition/identification rates decreases as the size/coverage of occlusion area increases.

4.3.2 Comparison of Results with those of Existing Approaches

The performance of the other occlusion robust approaches is validated in literature using accuracy score (Song et al., 2019), (Iliadis et al., 2017) and receiver operating characteristic (ROC) (Liao, et al., 2013) as the evaluation metrics. Therefore, to compare the performance of our approach to those in literature we used the accuracy

score and ROC curve. The performance of our approach to those in literature is shown in Table 4.8.

Table 4.8: Performance comparison of various approaches to ours

Dataset	Model	Accuracy (%)
FaceScrub	PDSN (Song et al, 2019)	74.40
	Detection and Exclusion of Occluded Face Regions Approach	**92**
Yale B	Illiadis et al,2017	95.82
	Detection and Exclusion of Occluded Face Regions Approach	**96**
LFW	PDSN (Song et al, 2019)	99.20
	Detection and Exclusion of Occluded Face Regions Approach	**92**

The results in Table 4.8 show that our approach performed highly compared to the state-of-the-art model on the FaceScrub and Yale B datasets. For (Song et al., 2019), they synthesized the FaceScrub dataset using different objects like sunglasses, book, phone. On the other hand, the FaceScrub dataset adopted for this research used masks as the only occlusion synthesis. Unlike (Song et al., 2019), our approach did not require occluded face images as part of the training set. Song et al. (2019), used the whole dataset for their experiment whereas we only used 20 classes. The high performance of our approach could be attributed to the exclusion/discarding of occluded regions before performing recognition. The use of non-occluded faces for training meant that the weakness of requiring all variation of occlusions to be used was mitigated.

For the Yale B dataset, the performance of our approach to that of (Iliadis et al., 2017) was almost similar. Our approach performed slightly higher than theirs with 0.18% increase as shown in Table 4.8. Iliadis et al. (2017) performed their experiments using block occlusions on the Yale B dataset whereas we synthesized the Yale B dataset using masks. Iliadis et al. (2017), used block occlusions at 60% whereas our synthesized mask was approximately at 50%. Our approach performed slightly better due to the discarding of the occluded regions before recognition.

The LFW dataset is used as a benchmark dataset in verifying Face recognition systems. There are many ways one can use them. One is doing the verification by identifying the images in the dataset and another way is using the LFW validation protocol (Huang et al., 2007) which was used in these experiments. From the Table 4.8 above, it is clearly shown that our approach performed lower in comparison to (Song et al., 2019). This could be attributed to the use of transfer learning for our model as compared to theirs in which they trained their model using 0.49 million training data. Despite the lower performance, the approach is still useful for performing face recognition with partial occlusions.

For Liao et al. (2013), they conducted an experiment 83 face images from 83 subjects plus the 5,000 images of 5,000 images from LFW dataset (Huang et al., 2007). On our part we used at most 20 images from 20 subjects for training and at most 10 images for testing from the Pubfig dataset (Kumar et al., 2009).

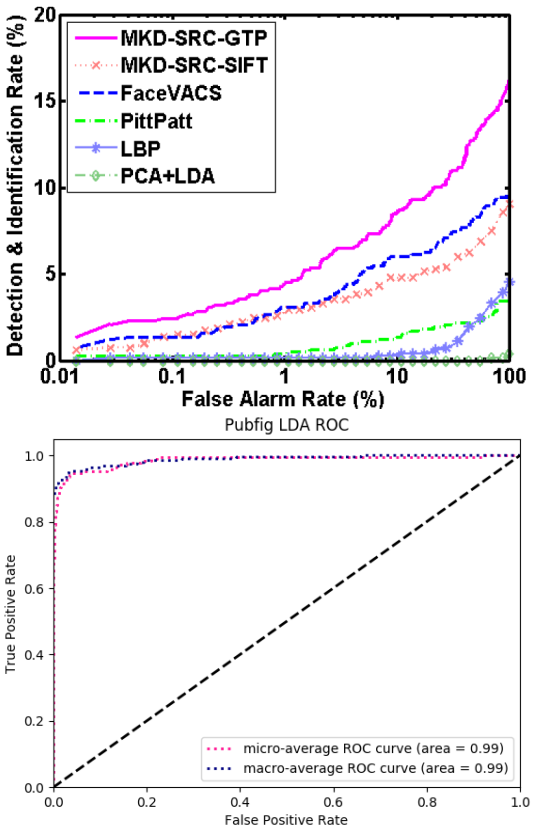


Figure 4.5: ROC curves from the Pubfig dataset (Kumar et al, 2009) from (Liao et al., 2013)

Figure 4.6: ROC curves from the Pubfig dataset (Kumar et al., 2009) our experiments

From Figure 4.5 and 4.6 it can clearly be shown that both the approaches perform well. However, it was challenging to compare the performance of these approaches because they all had different set up in terms of data and how that data was used to run the experiments. The Webface-OCC (Huang et al., 2021) was a new benchmark dataset and its validation protocol could not be adopted for this research.

4.4 Effect of the Number of Images in Learning a Class

To study the number of images required to learning a class effectively; experiments were run on a subset of the Pubfig dataset. The experiments involved running it non augmented data 0, data that had 5, 10 and 15 augmentations. A slight improvement in

the recognition rate was observed as the number of images increased as shown in Figure 4.7.

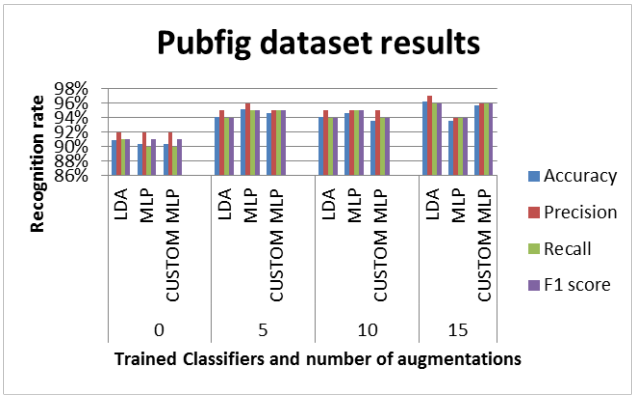


Figure 4.7: Chart showing the effect of number of images on recognition rate

4.5 Limitations of the Research

There was lack of adequate computing resources in terms of random-access memory therefore, the sample size used in running these experiments was small. Secondly, despite the efforts put in to acquire the real world occluded and common dataset among the state-of-art approaches, the AR dataset, for use in this research, it was not accessible.

Thirdly, synthetically occluded images with realistic occlusions such as masks, clothes, sunglasses were used to test the model. To study the occlusion effect on recognition rates, a random black rectangle was used. Therefore, this research did not cover other artificial occlusions such as random objects, for example; cats, cups and hands on the face images.

Finally, comparing this approach to the state-of-art models was challenging due to limited number and accessibility of standard benchmark datasets with partial occlusions. Additionally, every state-of-art approach used different datasets with different synthesized occlusions, training and validation protocols.

4.6 Research Outputs

i. The data used for this approach reflected real world occlusions and the approach performed fairly compared to the state-of-the-art approaches.
ii. It used less data in training as compared to deep learning models that require a lot of data to perform better, therefore, the approach can be used in real world applications.
iii. The use of non-occluded faces for training meant that the weakness of requiring all variation of occlusions to be used was mitigated.
iv. As the approach performed well in terms of accuracy, precision, recall and F1 score it can be integrated into criminal identification systems because; criminals have a tendency to hide part of their faces when committing crimes, therefore the will identified.
v. A new benchmark dataset, Webface-OCC was used in this research was used and the results were comparative to the state-of-art face recognition approaches robust to partial occlusions as it was close to the perfect score of 100%.

4.7 Summary

The chapter looked into the experiments' results and the derived detection and exclusion of occluded face regions approach. The effect of occlusion on face recognition/identification rates was also observed in this research. The performance of the approach was evaluated and its results were compared to those existing in literature. Finally, the limitations of conducting this research and the research outputs were outlined.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND FUTURE WORK

5.1 Introduction

This chapter briefly highlights this research's summary, conclusions and recommendation for future work.

5.2 Research Summary

Partial face occlusions such as scarfs, masks and sunglasses compromise face recognition accuracy. This research was aimed at developing a face recognition approach robust to partial occlusions. Based on the findings in chapter two an approach from existing approaches was developed as defined in the methodology section in chapter three. The developed approach was based on the assumption that the human visual system ignores occlusion and solely focuses on the non-occluded sections for recognition. It was an occlusion aware approach; therefore, the occluded parts were discarded during recognition. The detection and exclusion of occluded regions adopted two methods; use of haar cascade classifiers and use of skin detection.

The derived approach was based on an occlusion aware approach. The least occluded region of the face was retrieved from a partially occluded face and used for recognition. The final approach used the Inception Resnet V1 as a feature extractor, haar cascades as the least occluded region retrieval technique, LDA as both a dimensionality reduction algorithm and a classifier.

The performance of the face recognition approach to partial occlusions developed was evaluated as was done in chapter four by using measures of accuracy, precision, recall and F1 score. The approach performed relatively well in the classification task

with an accuracy of 92% on the Webface-OCC, 96% on the Pubfig, 92% on the FaceScrub, 96% on the Yale B and 92% on the LFW datasets.

5.3 Conclusions

In this research a face recognition approach that is robust to partial occlusions was developed. The effect of occlusion on performance of face recognition was also observed in this research and it showed that performance decreases as occlusion increases. The approach mitigated some of the state-of-the-art weaknesses such use of large volumes of data, where our approach only required at most 20 samples from each for training. Secondly, other approaches did not use datasets that reflected real world occlusion scenarios, our approach used the new synthetically occluded Webface-OCC dataset and an algorithm was used to synthetically occlude the Pubfig, FaceScrub and Yale B datasets by drawing masks on the face images to reflect the real-world scenario.

Finally, the problem that all types of occlusion variations for better performance was mitigated as the approach did not require occluded faces for training; it used non-occluded/whole face images. Additionally, such an approach's algorithm can be used in criminal identification systems because; criminals have a tendency to hide part of their faces when committing crimes.

5.4 Future Work

For future work, we plan to investigate if the approach is robust to scaling in terms of increasing the number of classes or faces. We will also perform an analysis on its computational complexity. Additionally, we will investigate on how to improve recognition rates on the sections or regions of the face like the mouth and nose.

On the other hand, we recommend that further research should be conducted on occlusion detection especially in the areas of skin detection and the improvement of haar cascade classifiers. The recognition rates dropped significantly beyond 60% occlusion coverage using this approach. Further research, should therefore be conducted to improve recognition rates beyond 60%occlusion coverage area. More

experiments should be conducted on the new benchmark dataset (Webface-OCC) for performance comparison and improvement on the results obtained. Finally, more datasets with real world occlusions should be developed.

REFERENCES

Alexey, AB. (2015). OpenCV Detection Models. Retrieved from https://github.com/AlexeyAB/OpenCV-detection-models/tree/master/haarcascades

Bansal, A. (2018). A study of factors affecting face recognition systems. *International Journal of Management and Engineering,8*. Retrieved from http://www.ijamtes.org/gallery/32.%20feb%20ijamtes%20-%20276.pdf

Bento, C. (2021). Multilayer Perceptron explained with a real-life example and python code: Sentiment analysis. *Towards Data Science*. Retrieved from https://towardsdatascience. com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141.

Bernstein, C. (2020). face detection. Retrieved from https://www.techtarget.com/searchenterpriseai/definition/face-detection

Brownlee, J. (2016). *Crash Course on Multi-Layer Perceptron Neural Networks.* Retrieved from https://machinelearningmastery.com/neural-networks-crash-course/

Cen, F., & Wang, G. (2019). Dictionary representation of deep features for occlusion-robust face recognition. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 26595–26605. doi:10.1109/access.2019.2901376

Cen, K. (N.D). *Study of Viola Real Time Face Detector*. Retrieved from https://web.stanford.edu/class/cs231a/prev_projects_2016/cs231a_final_report.pdf

Chen, X., Qing, L., He, X., Su, J., & Peng, Y. (2018). From eyes to face synthesis: A new approach for human-centered smart surveillance. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 14567–14575. doi:10.1109/access.2018.2803787

Dvornik, N., Mairal, J., & Schmid, C. (2019). On the importance of visual context for data augmentation in scene understanding. *IEEE transactions on pattern analysis and machine intelligence*, *43*(6), 2014-2028.

Dwivedi, D. (2018). Face recognition for Beginners. Retrieved from https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2.

Educative Answers Team (N/A). What is a multi-layered perceptron? Retrieved from https://www.educative.io/answers/what-is-a-multi-layered-perceptron.

Esler, T. (2021). Facenet Pytorch : Pretrained Pytorch face detection and recognition models. Retrieved from https://pypi.org/project/facenet-pytorch/

Etemad, K., & Chellappa, R. (1997). Discriminant analysis for recognition of human face images. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1206*(8), 125–142

Feng, Y., Yu, S., Peng, H., Li, Y. & Zhang, J. (2022). Detect Faces Efficiently: A Survey and Evaluations. *IEEE Transactions on Biometrics, Behaviour and Identity Science, 4* (1), 1-18.

Gao, Y., Xiong, N., Yu, W. & Lee H. (2019). Learning Identity-aware face features across poses based on deep Siamese networks. *IEEE Access: Digital Object Identifier 10.1109/ACCESS.2019.2932760*

Georghiades, A., Belhumeur, P. & Kriegman, D. (2001). From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. PAMI, 2001.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep residual Learning for Image Recognition. Retrieved from https://arxiv.org/abs/1512.03385

Huang, B., Wang, Z., Wang, G., Jiang, K., Zeng, K., Han, Z., Tian, X., & Yang, Y. (2021). When face recognition meets occlusion: A new benchmark. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2021-June*, 4240–4244.

Huang, G., Ramesh, M., Berg, T. & Learned-Miller, E. (2007). Labelled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49.

IBM Cloud Education (2021) Overfitting. Retrieved from https://www.ibm.com/cloud/learn/overfitting

Iliadis, M., Wang, H., Molina, R., & Katsaggelos, A. K. (2017). Robust and Low-Rank Representation for Fast Face Identification with Occlusions. *IEEE Transactions on Image Processing*, *26*(5), 2203–2218.

Ismail, N., & Sabri, M. I. M. (2009): Review of existing algorithms for face detection and recognition. *Proceedings of the 8th WSEAS International …*, *September*, 30–39.

Jean (2018). Skin Detection Algorithm. Retrieved from https://github.com/Jeanvit/PySkinDetection

Jetbrains, (N/A), Pycharm. Retrieved from https://www.jetbrains.com/pycharm/

Jia, H., & Martinez, A. M. (2008). Face recognition with occlusions in the training and testing sets. *2008 8th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2008*, 1-6.

Jung, A. (2021), imgaug. Retrieved from https://imgaug.readthedocs.io/en/latest/

Kakkar, P. & Sharma, V. (2018) Criminal Identification System using face detection and recognition. *International Journal of Advanced Research in Computer and Communications Engineering.*

Keil, M. (2009). "I look in your Eyes, Honey": Internal Face Features Induce Spatial Frequency Preference for Human Face Processing. *PLoS Computational Biology*, 2009; 5 (3): e1000329 DOI: 10.1371/journal.pcbi.1000329

Khan, M., Harous, S., Hassan, S., Khan, M., Iqbal, R. & Mumtaz, S. (2019) Deep unified model for face recognition based on convolution neural network and Edge computing. IEEE Access: *Digital Object Identifier 10.1109/ACCESS.2019.2918275*

Khandelwal, R. (2018). Convolutional neural networks simplified: Retrieved from *https://medium.com/datadriveninvestor/convolutional-neural-network-cnn-simplified-ecafd4ee52c5*

King, D. (2021) Dlib: A Toolkit for Making Real World Machine Learning and Data Analysis Applications. Retrieved from https://pypi.org/project/dlib/

Kolkur, S., Kalbandez, D., Shimpi, P., Bapat, C. & Jatakia, J. (2017). Human Skin Detection Using RGB, HSV and YCbCr Color Models. Retrieved from https://doi.org/10.48550/arXiv.1708.02694

Kong, J., Chen, M., Jiang, M., Sun, J. & Hou, J. (2018). Face Recognition Based on CSGF(2D)^{2}PCANet. *IEEE Access: Digital Object Identifier 10.1109/ACCESS.2018.2865425*

Kumar, N., Berg, A., Belhumeur, P. & Nayar, S. (2009). Attribute and Simile Classifiers for Face Verification. *International Conference on Computer Vision (ICCV).*

Kumar, S., Singh, S. & Kumar, J. (2017). A study on Face Recognition Techniques with Age and Gender Classification. *Conference Paper of May 2017.* Retrieved fromhttps://www.researchgate.net/publication/318348766

Li, L., Mu, X., Li, S. & Peng, H. (2020). A Review of Face Recognition Technology. *IEEE Access*. *Digital Object Identifier 10.1109/ACCESS.2020.3011028*

Li, S., Dou, Y., Xu, J., Yang, K. & Li, R. (2019) GBCNN: A full GPU-Based Batch Multi-Task Cascaded CNN. *IEEE Access: Digital Object Identifier 10.1109/ACCESS.2019.2894589*

Liao, S., Jain, A. & Li, S. (2013). Partial Face Recognition: Alignment-Free Approach. *IEEE Transactions on pattern analysis and machine intelligence, 35(5),* 1193–1205.

Mahdi, F., Habib, M., Vasant, P., Mckeever, S. (2017). Face recognition based real-time system for surveillance. *Intelligent Decision Technologies 11 (2017) 79–92 Research gate.*

Mao, L., Sheng, F., & Zhang, T. (2019). Face Occlusion Recognition with Deep Learning in Security Framework for the IoT. *IEEE Access*, *7*, 174531–174540.

Mein, W. (2020). Facial mask overlay with OpenCV-dlib. Retrieved from https://medium.com/mlearning-ai/facial-mask-overlay-with-opencv-dlib-4d948964cc4d

Min, R., Hadid, A. & Dugelay, J. (2014). Efficient Detection of Occlusion prior to Robust Face Recognition. *The Scientific World Journal*, vol. 2014. https://doi.org/10.1155/2014/519158

Mittal, A. (2020). Haar Cascades, Explained. Retrieved from https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d last

Ng, H. & Winkler, S. (2014). A data-driven approach to cleaning large face datasets. *Proc. IEEE International Conference on Image Processing (ICIP),* Paris, France, Oct. 27-30, 2014.

OpenCv(N/A). Image Segmentation with Watershed Algorithm Retrieved from https://docs.opencv.org/4.x/d3/db4/tutorial_py_watershed.html.

Parkhi, O., Vedaldi, A. & Zisserman, A. (2014). Deep Face Recognition.

Pawangfg (2021). ML | Face Recognition Using Eigenfaces (PCA Algorithm). Retrieved from https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., and Blondel, M. and Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *JMLR* 12, pp. 2825-2830.

Python, (N/A), Python. Retrieved from https://www.python.org/.

Qi, R., Jia, R., Mao, Q., Sun, H. & Zuo, L. (2019) Face Detection Method Based on Cascaded Convolutional Networks. *IEEE Access: Digital Object Identifier 10.1109/ACCESS.2019.2934563*

Ray (2020). Computer Vision – Watershed Algorithm. Retrieved from https://medium.com/analytics-vidhya/computer-vision-watershed-algorithm-ca16bd00485

Razman, F., Khan, M., Rehmat, A., Iqbal, S., Saba, T., Rehman, A. & Mehmood, Z. (2019). A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. *Journal of medical imaging*. Retrieved from https://doi.org/10.1007/s10916-019-1475-2

Rosebrock, A. (2021). Imutils: A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3. Retrieved from https://pypi.org/project/imutils/

Saini, Y. (2022). Various Techniques used for Face Recognition. Retriebed from https://iq.opengenus.org/techniques-for-face-recognition/

Santonkar, S., Kurhe B. &Khanale, B. (2011). Challenges in Face Recognition: A Review. *International Journal of Advanced Research in Computer Science, 2*(4).

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June-2015*, 815–823.

Simonyan, K. & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from https://arxiv.org/abs/1409.1556

Sirovich, L. & Kirby, M. (1986). Low-dimensional procedure for the characterization of human faces. *Journal of the optical society of America A, Vol. 4, page 519.*

Song, L., Gong, Di., Li, Z., Liu, C., & Liu, W. (2019). Occlusion robust face recognition based on mask learning with pairwise differential siamese

network. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-October*, 773–782.

Sun, Y., Wang, X. & Tang, X. (2014). Deep Learning Face Representation from Predicting 10,000 Classes. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*

Syafeeza, A., Khalil-Hani, M., Liew, S. & Bakhteri, R. (2014). Convolutional Neural Network for Face Recognition with Pose and Illumination Variation. *International Journal of Engineering and Technology (IJET).*

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Retirved from https://arxiv.org/pdf/1602.07261.pdf

Tarrase, M. (2018). What is wrong with Convolutional neural neworks? Retrieved from https://towardsdatascience.com/what-is-wrong-with-convolutional-neural-networks-75c2ba8fbd6f.

Torch Contributors (2022). Retrieved from https://pytorch.org/vision/stable/index.html.

Towner, H. & Slater, M. (2007). Reconstruction and Recognition of Occluded Facial Expressions Using PCA. In: Paiva, A.C.R., Prada, R., Picard, R.W. (eds) Affective Computing and Intelligent Interaction. *ACII 2007*. Lecture Notes in Computer Science, vol 4738. Springer, Berlin, Heidelberg. Retrieved from https://doi.org/10.1007/978-3-540-74889-2_4

Turk, M. & Pentland, A. (1991). Face Recognition Using Eigenfaces. *Vision of Modelling Group, The Media Laboratory MIT.*

Tyagi, M. (2021). HOG (Histogram of Oriented Gradients): An Overview. https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Vijayalakshmi, A. (2017). Recognizing Faces with Partial Occlusion Using Inpainting. *International Journal of Computer Applications,168(13),* 20-24.

Viola, P. & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. Retrieved from https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

Wang, M., Hu, Z., Sun, Z., Zhao, S., & Sun, M. (2017). Varying face occlusion detection and iterative recovery for face recognition. *Journal of Electronic Imaging*, *26*(3), 033009.

Wang, W., & Carreira-Perpiñán, M. (2014): The role of dimensionality reduction in classification. *Proceedings of the National Conference on Artificial Intelligence*, *3*(2), 2128–2134.

Wanyonyi, D. & Celik, T. (2022). Open-Source Face Recognition Frameworks: A Review of the Landscape. IEEE Access. *Digital Object Identifier 10.1109/ACCESS.2022.3170037*

Wei, H., Lu, P. & Wei, Y. (2020). Balanced Alignment for Face Recognition: A Joint Learning Approach. Megvii Technology, Fudan University. arXiv:2003.10168v1

Wei, X., Li, C., Lei, Z., Yi, D. & Li, S. (2014). Dynamic Image-to-Class Warping for occluded Face Recognition. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 12.* pp 2035-2050

Wen, Y., Liu, W., Yang, M., Fu, Y., Xiang, Y. & Hu, R. (2015). Structured occlusion coding for robust face recognition. Retrieved from https://doi.org/10.48550/arXiv.1502.00478

Wu, C. Y., & Ding, J. J. (2018): Occluded face recognition using low-rank regression with generalized gradient direction. *Pattern Recognition*, *80*, 256–268.

Wu, J. (2017). Introduction to Convolutional Neural Networks. *National Key Lab for Novel Software Technology Nanjing University, China.* Retrieved from https://pdfs.semanticscholar.org/450c/a19932fcef1ca6d0442cbf52fec38fb9d1e5.pdf

Xiao, J. Li, S. & Xu, Q. (2019). Video based evidence analysis and extraction in digital forensic investigation. *IEEE Access: Special section on deep learning: security and forensics research advances and challenges*. Digital Object Identifier 0.1109/ACCESS.2019.2913648

Xiaozhou, Y. (2020). Linear Discriminant Analysis. Retrieved from https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b.

Yang, M., Wang, X., Zeng, G., & Shen, L. (2017). Joint and collaborative representation with local adaptive convolution feature for face recognition with single sample per person. *Pattern Recognition*, *66*(December 2016), 117–128

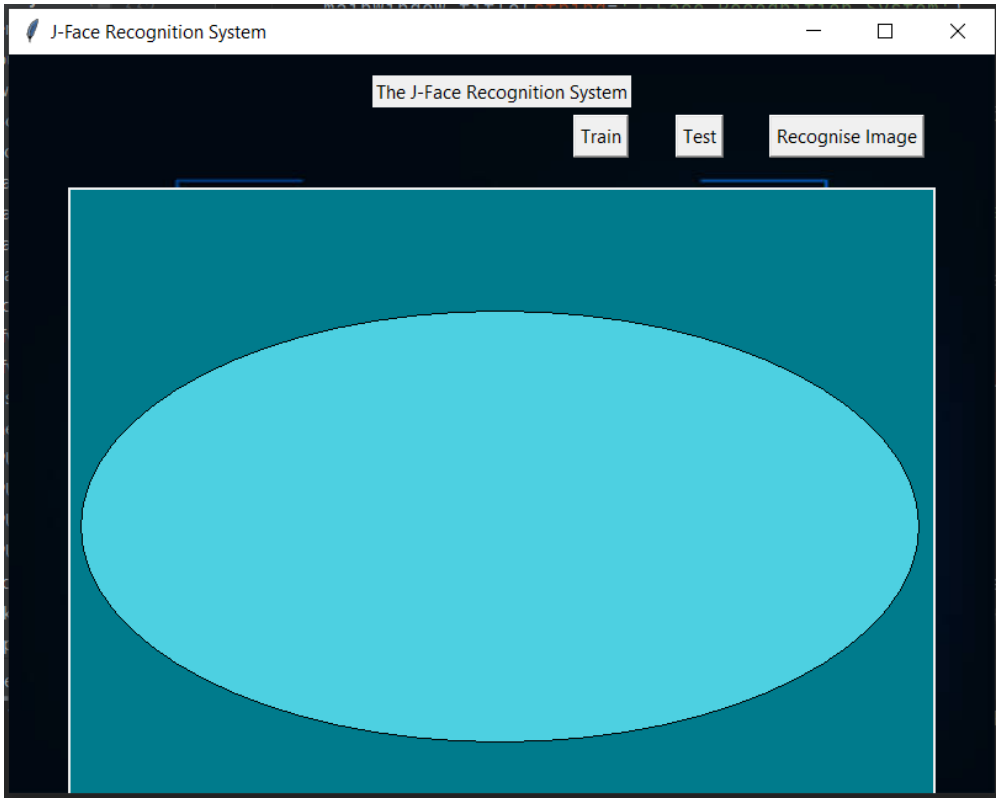
Zeng, D., Veldhuis, R., & Spreeuwers, L. (2021): A survey of face recognition techniques under occlusion. *IET Biometrics*. https://doi.org/10.1049/bme2.12029

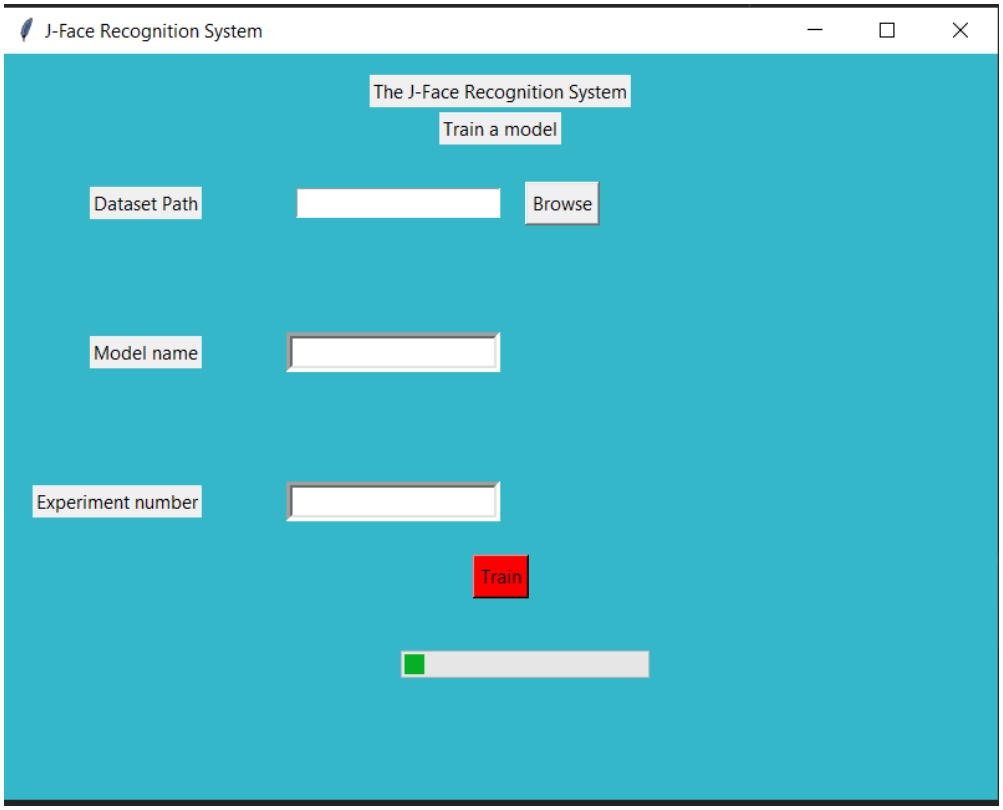
Zhang, L., Verma, B., Tjondronegoro, D. & Chandran, V. (2018). Facial Expression Analysis Under Partial Occlusion: A Survey. Retrieved from https://doi.org/10.48550/arXiv.1802.08784

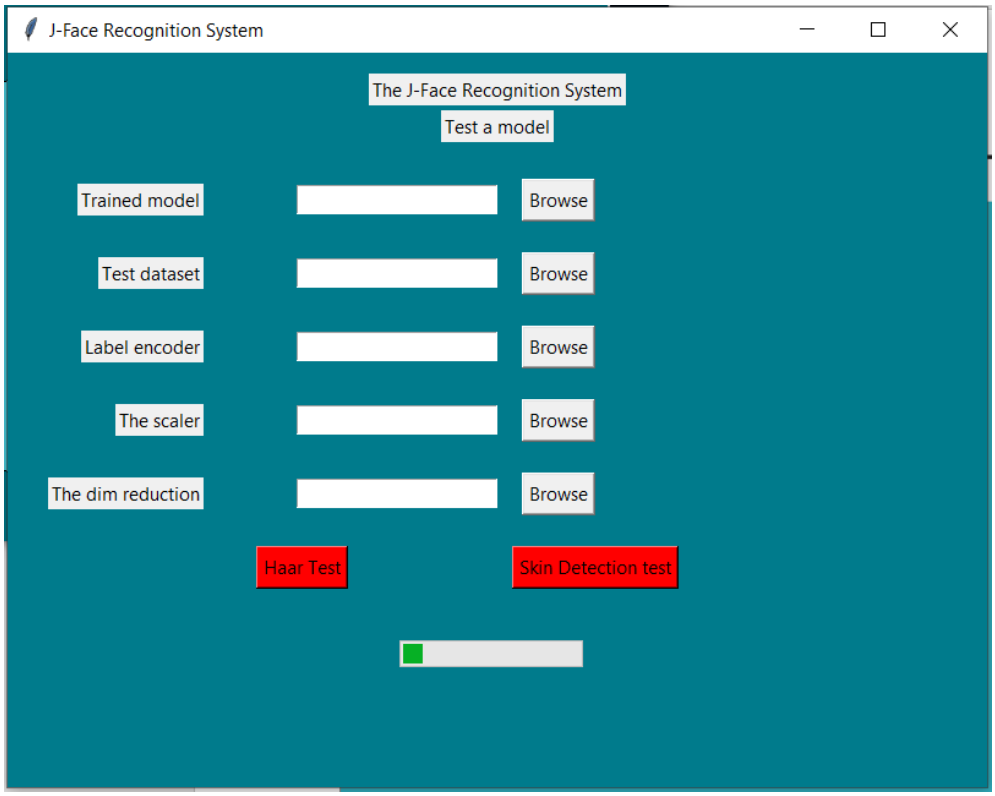
Zhou, E., Cao, Z. & Yin, Q. (2015). Naive-Deep Face Recognition: Touching the Limit of Lfw Benchmark or Not? arXiv preprint arXiv:1501.04690.

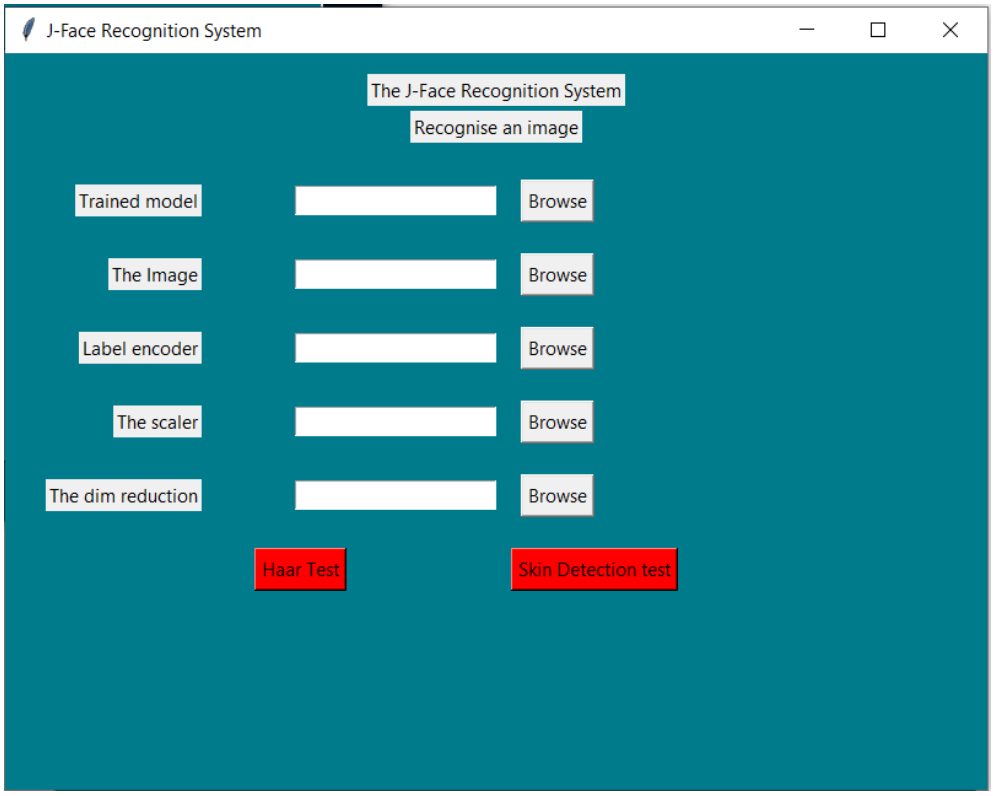
APPENDICES

Appendix I: The Graphical User Interface




J-Face Recognition System
The J-Face Recognition System
Train a model
Dataset Path
Browse
Model name
Experiment number
Train


J-Face Recognition System
The J-Face Recognition System
Test a model
Trained model
Browse
Test dataset
Browse
Label encoder
Browse
The scaler
Browse
The dim reduction
Browse
Haar Test
Skin Detection test


J-Face Recognition System
The J-Face Recognition System
Recognise an image
Trained model
Browse
The Image
Browse
Label encoder
Browse
The scaler
Browse
The dim reduction
Browse
Haar Test
Skin Detection test

Appendix II: Sample codes

```
#pretrained model

facenet = InceptionResnetV1(pretrained='vggface2').eval()

new_model=facenet
device = torch.device('cuda:0' if torch.cuda.is_available() else
"cpu")
new_model = new_model.to(device)



#classifiers

lda = LinearDiscriminantAnalysis()
mlp =OneVsRestClassifier(MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=(5000,), random_state=0, max_iter=2500,
                   learning_rate_init=0.0008, tol=1e-2))


#derive face sections/regions



def divideImages(img):
    height, width, channels = img.shape
    width_cutoff = width // 2
    s1 = img[:, :width_cutoff]
    s2 = img[:, width_cutoff:]

    height_cutoff = height // 2
    s3 = img[:height_cutoff, ]
    s4 = img[height_cutoff:, ]

    return s1, s2, s3, s4



#skin detection

def detskin2(img):
    detector = skinDetector(img)
    theoutput = detector.find_skin()
    return theoutput
```

```
#feature extraction


def featureextractionVgg(img):
    img = transform(img)
    img = img.reshape(1, 3, 160, 160)
    # extract features, don't need gradient
    with torch.no_grad():
        # Extract the feature from the image
        feature = new_model(img)
        # Convert to np array, reshape and save it to a variable
        feature = feature.cpu().detach().numpy().reshape(-1)
        return feature



#feature scaling



def scaletrainencodings(the_scaler, the_encodings):
    the_scaler.fit(the_encodings)
    transformed_encodings = the_scaler.transform(the_encodings)
    return the_scaler, transformed_encodings

#model training



def model_training(the_face_labels,
the_face_encodings,the_classifier):
    face_encodings = np.asarray(the_face_encodings)
    face_labels = the_face_labels
    the_model = the_classifier.fit(face_encodings, face_labels)
    score = the_model.score(face_encodings, face_labels)
    return the_model,score



#model testing


def model_testing(the_model, test_encodings, test_labels):
    the_test_ncodings_val_std = np.asarray(test_encodings)
    test_labels = test_labels
    since = time.time()
    #the predictions = the model.predict(the test ncodings val std)
    the_score = the_model.score(the_test_ncodings_val_std,
test_labels)
    print("testing complete")
    return the_score
```

```
#haar features

downloaded_haars = 'OpenCV-detection-models-master/haarcascades/'
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_eye.xml')
mouth_cascade = cv2.CascadeClassifier(downloaded_haars +
'haarcascade_mcs_mouth.xml')
left_eye = cv2.CascadeClassifier(downloaded_haars +
'haarcascade_mcs_lefteye.xml')
right_eye = cv2.CascadeClassifier(downloaded_haars +
'haarcascade_mcs_righteye.xml')



#detect left eye

def detect_left_eye(gray, height):
    left_eye = left_eye.detectMultiScale(gray)
    if len(left_eye) > 0:
        for (ex, ey, ew, eh) in left_eye:
            if ey + eh > height / 2:
                return "no eyes"
            else:
                return "1"
    else:
        return "no eyes"



#detect right eye


def detect_right_eye(gray, height):
    right_eye = right_eye.detectMultiScale(gray)
    if len(right_eye) > 0:
        for (ex, ey, ew, eh) in right_eye:
            if ey + eh > height / 2:
                return "no eyes"
            else:
                return "2"
    else:
        return "no eyes"


#detect both eyes



def detect_eyes(gray, height):
    eyes = eye_cascade.detectMultiScale(gray)
    if len(eyes) > 0:
        for (ex, ey, ew, eh) in eyes:
            if ey + eh > height / 2:
```

```
                return "no eyes"
            else:
                return "3"
    else:
        return "no eyes"

#detect mouth


def detect_mouth(gray, height):
    eyes = mouth_cascade.detectMultiScale(gray, 1.2, 5)
    if len(eyes) > 0:
        for (ex, ey, ew, eh) in eyes:
            if ey + eh < height / 2:
                return "no mouth"
            else:
                return "4"
    else:
        return "no mouth"


def select_region(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    height = np.size(img, 0)
    eyes = detect_eyes(gray, height)
    if eyes == "3":
        return "3"
    else:
        mouth = detect_mouth(gray, height)
        if mouth == "4":
            return "4"
        else:
            left_eye = detect_left_eye(gray, height)
            if left_eye == "1":
                return "1"
            else:
                right_eye = detect_right_eye(gray, height)
                if right_eye == "2":
                    return "2"
                else:
                    return "hello"

def display_image(img,thelabel):
    detector = dlib.get_frontal_face_detector()
    input_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_h, img_w, _ = np.shape(input_img)

    detected = detector(input_img, 1)

    for i, d in enumerate(detected):
        x1, y1, x2, y2, w, h = d.left(), d.top(), d.right() + 1,
d.bottom() + 1, d.width(), d.height()
        cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 0), 2)

    cv2.imshow(thelabel, img)
    cv2.waitKey()
```