Face Recognition using Fisherface and PCA with CNN

Balasubramaniyam Muthuramalingam

M.Sc. in Computing
in Big Data Analytics
and AI

2021

Computing Department, Letterkenny Institute of Technology, Port Road, Letterkenny, Co. Donegal, Ireland.

Face Recognition using Fisherface and PCA with CNN

Author: Balasubramaniyam Muthuramalingam

Supervised by: Gary Cullen

A thesis submitted in partial fulfilment of the requirements for the

Master of Science in Computing in Big Data Analytics and AI

Submitted to Quality and Qualifications Ireland (QQI)
Dearbhú Cáilíochta agus Cáilíochtaí Éireann

September 2021

Declaration

I hereby certify that the material, which I now submit for assessment on the programmes of study leading to the award of Master of Science in Computing in Big Data Analytics and Artificial Intelligence, is entirely my own work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of my own work. No portion of the work contained in this thesis has been submitted in support of an application for another degree or qualification to this or any other institution. I understand that it is my responsibility to ensure that I have adhered to LYIT's rules and regulations.

I hereby certify that the material on which I have relied on for the purpose of my assessment is not deemed as personal data under the GDPR Regulations. Personal data is any data from living people that can be identified. Any personal data used for the purpose of my assessment has been psudonymised and the data set and identifiers are not held by LYIT. Alternatively, personal data has been anonymised in line with the Data Protection Commissioners Guidelines on Anonymisation.

I consent that my work will be held for the purposes of education assistance to future students and will be shared on the LYIT Computing website (www.lyitcomputing.com) and Research THEA website (https://research.thea.ie/). I understand that documents once uploaded onto the website can be viewed throughout the world and not just in the Ireland. Consent can be withdrawn for the publishing of material online by emailing Thomas Dowling; Head of Department at thomas.dowling@lyit.ie to remove items from the LYIT Computing website and by email emailing Denise McCaul; Systems Librarian at denise.mccaul@lyit.ie to remove items from the Research THEA website. Material will continue to appear in printed formats once published and as websites are public medium, LYIT cannot guarantee that the material has not been saved or downloaded.

Balasubramaniyam M	30th August 2021
Signature of Candidate	Date

Acknowledgements

I would like to thank Letterkenny Institute of Technology and my professors, Dr Shagufta Henna, Dr Kevin Meehan, Dr Mark Leeney and Edwina Sweeney for guiding me throughout the academic year.

I am grateful to my supervisor, Dr Gary Cullen, for his critical guidance and providing motivation throughout the dissertation.

I am very thankful to my mother and father: Saraswathi and Muthuramalingam for supporting me. Lastly, I would like to thank my brother and friends: Mr Sai Karthik, Mr Abhiram Gopal and Ms Michelle Yong for their constant support.

Abstract

Face Recognition has evolved over the past few years, the primary focus of this technology is to enhance security by updating the existing biometric scanners. Because of its popularity, most of the industries have shifted their focus to improve this tool. One of the primary problems of face recognition is providing unauthorised access due to false identification. The factor lies in different section, poor accuracy, lighting, alignment or facial expression. Due to rapid development in Artificial Intelligence (AI), many researchers have contributed their time and resources to produce different solutions for such problems. A face recognition model detects a face and analyses for facial features to be extracted. This face print is stored for identification. In this research, the features are extracted by using two face algorithms, Principal Component Analysis (PCA) and Fisherface. Both the algorithms use dimensionality reduction approach, where the image size is reduced to its simplest form for faster performance in cost of accuracy. The facial feature extracted from each algorithm is stored separately for performance comparison. A relatively new technique called Convolutional Neural Network (CNN) is used as a baseline to train the model. PCA and Fisherface are primitive face algorithms compared to CNN, so this paper highlights the performance of Fisherface and PCA when coupled with CNN. The experimental results showed that Fisherface performs better when coupled with CNN compared to PCA by achieving an accuracy of 67%.

Acronyms

Acronym	Definition	Page
AI	Artificial Intelligence	1
PCA	Principal Component Analysis	3
CNN	Convolutional Neural Network	3
CV	Computer Vision	3
ML	Machine Learning	9
DFM	Dynamic Feature Matching	15
FCN	Fully Convolutional Neural Network	15
GAP	Global Average Pooling	17
FLD	Fisher Linear Discriminant	17
LBP	Local Binary Pattern	17
FR	Face Recognition	18
PAW	Piecewise Affine Wrapping	20
LDA	Linear Discriminant Analysis	21
LPP	Local Preserving Projection	21
FD	Fourier Descriptors	23
LD	Local Descriptors	24
SIFT	Social Invariant Feature Transform	24
SLD	Sampling based Local Descriptors	24
CSV	Comma Separated Values	28
B&W	Black and White	28

Table of Contents

Table of Figures

Table of Tables

Table of Code Listings

1. Introduction

Face recognition is defined as identifying an individual's face by retrieving the matching face pattern stored in the database. It stores the unique facial features of the face mathematically and saves the data as face print (S. Gillis et al. 2019). Recent years, face recognition is becoming more common as it is being used in different fields like financial sectors, healthcare and law enforcement agencies (Sithara and Rajasree 2019) . For example, most hospitals in cities have implemented this technology to recognise previous patients and display their historical medical data stored. The system was made possible to implement because of rise of usage in internet, since many users are connected virtually, it was easier to access data of an individual and gather basic information about them. Face recognition is similar to face detection, both the concepts are different but carry similar process, face detection identifies a face in an image whereas face recognition simply identifies a particular(specific) face in a given image, provided that the specified faceprint had been registered in the system's database.

1.1. Purpose

Facial Recognition works as an identity tool which is widely utilised by major organisation to support their working environment. Initially, fingerprint scanners were used as modern biometric tool, eventually, this technology was simplified and installed in portable devices like mobile phones. This created a new path for developers to upgrade their biometric security by moving forward which led to retina scanner and face recognition. Similarly facial recognition will be established as a fundamental base in the future for contemporary discoveries. The plausibility of such event is very high considering the rapid growth in Artificial Intelligence (AI). Facial Recognition is an application that can reduce the probability of false identity, it provides an additional layer of security. An image provides more detailed information compared to transcript texts or vocal messages, hence analysing an image provides more content to the object. In terms of security, facial recognition reduces the probability of providing compromised environment. Since face is an ideal source of person's identity and plays an important role in conveying emotions, implementing this application will enhance the overall security of a device.

The current era is equipped with interactive technologies, most of the organisation and industry, have changed their way of work. For instance, in law enforcement, most of the areas are under surveillance, constantly watched especially near traffic junctions, this is to prevent from any motor vehicle to break the traffic law, if such event occurs, the surveillance camera would capture an image of the car along with the driver, this image will be further focused to reveal the person's image and the car's registration number, after a period of time the car's information is gathered[owner, registered county] and tracked down to verify if the driver's face matches with the owner to issue a ticket for violating the law (Frederike Kaltheuner' 2020).

1.2. Background

Computers have been proved to be more efficient and productive than humans but these machines require external assistance to operate them, to negate this predicament, many programmers and researchers have been focusing on creating a machine that has the capability to understand and learn new things, hence it was coined as Artificial Intelligence (AI).

Machine Learning is a subset of Artificial Intelligence, it means a machine trying to "learn" to complete a specific task. Through multiple iteration the machines can learn from experience and continue to expand their understanding and it is done by programming the model. The scripts written for the machine follows as set of instructions to guide through the entire process. The system's art of learning is similar to a human being, both the entities learn from previous experience. When the model is introduced to a dataset for testing, using the accumulated experience from the past it will be able to provide an appropriate result. The learning phase is called training the model where it tries to understand the algorithm used to define the model, more the iteration better the understanding. In Layman's term AI is defined as follows (Reynoso, 2021):

"AI is the concept where machines can interpret, mine and learn from external data in a way that functionally imitates cognitive behaviours traditionally associated with humans. It was fundamentally created on the grounds of human mental process which can be replicated and automated."

A face has various features, every person has different shapes of nose, lips and eyes. Each face has a different facial pattern, this information is extracted using Neural Networks. It is a subset of Deep Learning; it is widely used in image processing for feature extraction. According to an article on Face recognition using Neural Networks, ingenious Neural Network, features of the faces can be effectively extracted and achieve higher face recognition accuracy (Lu *et al.* 2021).

1.3. Problem Context

In the past, researchers used Eigenface Principal Component Analysis (PCA) for face recognition because of its fast computational speed. The cost of increasing the performance was to lose some accuracy because PCA periodically reduces the size of image by discarding irrelevant information. If the image contains facial expression, illumination or complex background, it fails to perform accurately. This problem can be tackled by limiting the dataset which contains images with simple facial features. But it can be avoided by using Fisherface algorithm which uses eigenface to calculate the Euclidian's distance.

The modern face recognition uses advanced technologies like Computer Vision (CV), Convolutional Neural Networks (CNN). Over the years, CNN is widely used for image classification. Hence, this paper tries to highlight the performance of Fisherface over PCA under CNN model.

1.4. Motivation

The concept of face recognition has not reached its full potential as it is under continuous development, many different applications have yet to produce consistent positive results while recognizing a face, when a face is not aligned or does not possess minimum detail required for the machine to identify the face it may lead to inaccurate results, but this hasn't affected the global market as the revenue keeps increasing yearly. The estimated amount in 2020 was around 3.8 billion US dollars and the market is projected to increase up to 8.5 billion

US dollars by 2025(Statista 2021). This proves that face recognition technology can control the future market filled with opportunities for developers to work on.

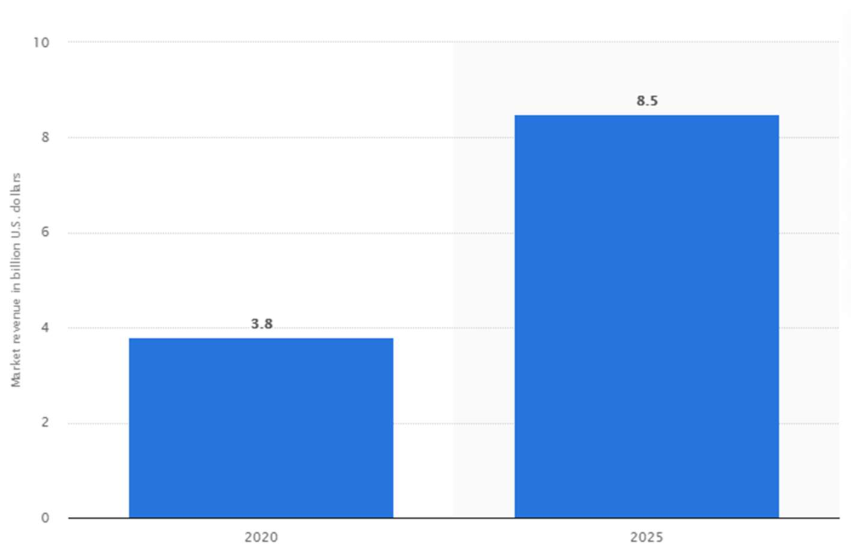

Figure 1 Facial Recognition global market size ('Statista ' 2021)

In the past, creating a bank account or applying for licence requires official document, like birth certificate or government ID and physical presence for verification. But now, all the information is stored in secure cloud and it can be accessed through internet. For instance, creating a driver licence is made easier using different types of biometric scanner, a person's identity can be verified and displaying all the relevant information to complete the process. The process has been simplified; time consumption has been drastically reduced. It was possible because the amount of time invested and contributions made by computer researchers. The evolution of Face Recognition in the past two decades displays the amount of interest shown by other developers and have realized, this technology is essential for the society.

1.5. Research Questions

This research intents to show a comparison of performance between Eigen Based Methods and Distribution Based Methods. Using Convolutional Neural Network as a baseline two different algorithms will be implemented from each method: PCA and Fisherface for feature extraction.

The research questions are given below:

Is it possible to determine the performance of Fisherface algorithm when coupled with Convolutional Neural Network?

Can CNN with PCA performance report be compared with CNN with Fisherface?

1.6. Report Outline

This chapter introduced the concept of face recognition and the evolution of the face recognition system throughout the years. A brief presentation on different methods and techniques used to enhance the performance of the model. Different applications and purpose of face recognition have been listed. The next chapter literature review briefs about the concepts involved in face recognition and related work accomplished by other researchers. Topics like Artificial Intelligence, Neural Networks and algorithms like Fisherface and PCA will be detailed.

2. Literature Review

2.1. Facial Recognition as a tool

As time progress, new discoveries are made, one of the prominent tools of 20^{th} Century is face recognition. Face recognition has been rapidly growing in popularity and gaining attention by computer vision scientist, it is aided by the availability of smartphones, laptop and tablets with cameras, furthermore, the image quality of these cameras has improved in recent years (Yohanes *et al.* 2018). Different economic sectors have inherited this technology for benefits, improve their work and social environment. It flourished in the biometric security category, there are other form of biometric software like voice recognition, fingerprint recognition and iris scanner. Face recognition is used in law enforcement, airports, healthcare, banking and other financial sectors, recently it has also been used as an additional security layer in mobile phone for unlocking the device. As a result, one must take precautions, when it affects privacy, concern about security, site vulnerability and legal requirements. Raise in worldwide concerns about terrorist activity and identity have fuelled the need for increased security. End users are more aware of potential of biometric technologies in security application as a result there's been increased media attention towards this tool (Sumathi and RaniHema Malini 2010).

2.1.1. Healthcare

E health has been growing rapidly in healthcare sector, especially during pandemic it has been proved to be quite useful to minimize contact and to control Covid-19, face recognition and AI are the forefront of making this achievable, an application has also been developed that allows to track down people who are on quarantine. Even in hospitals face recognitions is used to assist with patient care, the healthcare providers use it to access patients records and track their medicinal history, streamline patient registration, detect emotion and pain in patients. Most importantly, it also assists them in identification of specific genetic diseases (Sumathi and RaniHema Malini 2010). Individual identification is also one of the usages of this system, after comprehensive analyzation the following are required in healthcare for identity verification:

- Health insurance ID certificate (or govt. authenticated card).
- Online consulting services

- Immediate transportation
- Periodic Health Check-up
- Refund of medical fees
- Client record browsing

2.1.2. Law Enforcement

To create a civilized community, government implemented law enforcement but rapid increase in population, it is quite difficult to keep every area secure in a country. Along with rise in population, technology was also continuously moving forward, face recognition and its application have been able to show their true potential by fighting against crime and monitoring vulnerable areas of cities. It has become a regular tool among police agencies, the US government along with federal state and other enforcement agencies have agreed to invest in facial biometrics, it is estimated to increase from 136.9 million US dollars in 2018 to 375 million US dollars by 2025 (crime *et al.* 2021). With a portable device which supports face recognition system that is connected to law enforcement database provides an access to all the information of every criminal, hence in real time during patrol, officers can submit an image of random pedestrian and compare it with the database for identification, unlike DNA verification which may take up several days to get the results face recognition system proves to be productive. In 2019, Pew Research Centre conducted in US, if adults were ready to consent to use face recognition by law enforcement to ensure public safety. It is shown that over 59% of the participant agreed and 15% declined and the rest were not sure about the situation ('Statista' 2021).

Figure 2 US acceptance on Facial Recognition ('Statista' 2021)

Before this technology becomes available globally to cover every part of the world, flaws and other imperfections should be minimized before it creates any problems. To avoid such scenarios, imposing appropriate ethical laws and provide alternate procedure to control the scene.

2.1.3. Finance and Banking Sector

Many countries are still struggling with pandemic, as the lockdown impacted people's regular life. Use of online payment/transaction has surged over the years to avoid physical contact, if payments details were exposed to insecure website, it may be lead to compromised account. To avoid such situations, banking sectors have implemented additional security layer which can be manually enabled to maintain the integrity of the account. Biometric security ensures that authenticated users are allowed to access their account and reduces the probability of false identity. Passwords won't be compromised if bank payments or transactions are made using facial recognition. What if customer's face is compromised? Every modern facial application has a security feature built in, to check if the face is live human being or fake representation but this technique depends on accuracy and precision of the model. Recently, statistics showed a huge spike in the market of contactless biometric technologies, in 2020 it is observed to be around 8.25 billion US dollars but by 2027 it is estimated to reach 30.15 billion US dollars ('Global contactless biometrics market revenue 2027' 2021).

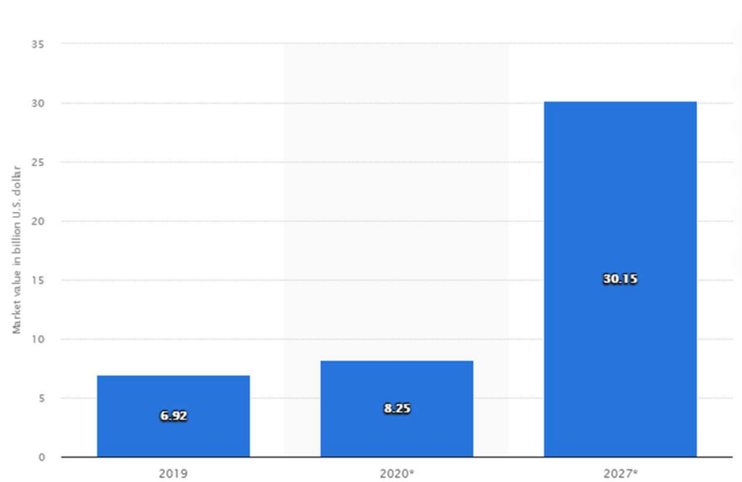

Figure 3 Contactless biometric technologies market size ('Statista' 2021)

2.2. Artificial Intelligence

Although most attempts to precisely define complex and widely used terms are futile, it is useful to draw an approximate boundary around the concept to provide context (Shivashankar B Nair et al. 2009). Artificial Intelligence (AI) is the study of how to program computers to perform tasks that humans currently perform better. It was officially introduced in 1956 by John McCarthy, a computer scientist. But the concept of AI was already familiar from 1400's, one of the greatest philosophers defined it as, "*There are three kinds of intelligence; one kind understands things for itself, the other appreciates what others can understand, the third understands neither for itself nor through others. This first kind is excellent, the second is good and the third kind useless*" (Niccolo Machiavelli, 1469-1527). AI provides systems that is able to resemble human perception, it helps in finding an optimal solution for the problem, if the required information is insufficient, it is able to solve with partial knowledge. It drops out some foundational properties of algorithm in order to obtain an acceptable solution (Ondrišová *et al.* 2019).

2.2.1. Convolutional Neural Networks

Convolutional neural network (CNN) is an idea developed from artificial neural network that was proposed when its unique network structure was discovered, which is capable of constructively decreasing the intricacy of feedback neural network. The idea was proposed by Hubel and Wiesel, who were examining the neurons used for local sensitivity and direction selection in the cat cerebral cortex, in the 1960s. concurrency, robustness, high fault tolerance, and nonlinearity are important features of the biological neural network, due to which the artificial neural network has found use in image processing, pattern recognition, and various other areas (Wang *et al.* 2020).

2.2.2. Machine Learning

Machine Learning (ML) is a subset of AI, the machine tries to understand the context of the problem by using different kinds of algorithm to provide an optimal result. Algorithms are trained by using statistical methods to make classifications and predictions, enables to find key insights about the projects. Following that, these insights drive decision-making within applications and businesses, with the goal of influencing key growth metrics (IBM, 2020).

There are various applications that uses ML as a baseline, for instance, Apple Siri's Speech Recognition, Apple's Face ID or Google Assistant. In simple context, assisting an infant to learn to walk or distinguish colours. There are two steps involved to achieve this goal, firstly by providing the system with a colour template appended with the name of each colour (data) to build a memory of colour. The second step is to repeat the first step, but the appended colour's name is removed. The system will try to distinguish the colour without any data support and will learn to describe each colour and identify the name from experience (training phase done in step one). There are many purposes for the use of Machine Learning and the depth of the concept has not been explored completely, each day there is a significant improvement in the field and the current researchers highly support the field of AI.

2.2.3. Steps in Face Recognition

An image is passed as an input to find facial features, this process is called face detection. Often non-frontal image detection is done to avoid performance problems. Recently deployed models on face recognition have been successful because the system uses dimensionality reduction techniques and subspace modelling. But finding such subspaces in an image in order to focus effective features and build robust classifiers is a very challenging task. Face recognition is dependent on high accuracy to achieve low intrusiveness, these key factors have drawn attention among researchers from various fields such as psychology to computer vision. Face detection in acquired images regardless of scale and location is the earliest stage. It relies on modern filters to differentiate point of region that shows faces and highlights them with robust classifiers. If the image requires change in attributes like scaling, translation and rotational variations it is done during the detection phase because facial expressions and alignment changes are major variations in early stages ('IntechOpen' 2021).

Newly developed facial recognition technologies utilise an anthropometric data set based system to predict approximate points of features found on the face. Estimated locations of the principal features on the face, such as the eyes, nose and mouth will first be determined (source). As the process is repeated, the system predicts sub-features on the face relative to its principal features. The entire procedure is checked and verified with collocation statistics to reject any detected points of mislocated (missing or misplaced?) features.

As the image of the face is analysed by the system, dedicated anchor points will be generated based on geometric combinations that can be identified. The process of facial recognition will then begin. The process requires the local representation of the facial appearance at each of the fixed points. The representation scheme will vary depending on the approach taken to create these recognition algorithms. To address and overcome predictable or unforeseeable complications, several algorithms have already been invented/developed/established in an attempt to develop a true constant, unchanging algorithm for recognition.

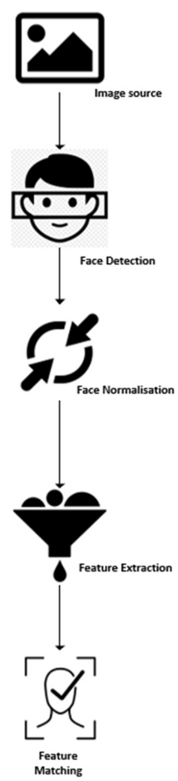

Figure 4 Face Recognition Process

Facial recognition technology is still very much a work in progress and has its limitations. Performance depends heavily on lighting conditions, facial expressions, visibility of the subject (due to video resolution or distance). Where all these factors are compromised, so does the technology's performance. Facial recognition technology and its algorithms are still far from perfection and only promises performance in controlled environments or when used to process media that meet the ideal requirements needed for a clear analysis. There is still a question of how to effectively collect and access facial templates generated from recognition technology. Facial features are extracted from images or videos and saved as a set of combinations or patterns. However, facial recognition remains to be a complex process and its technology is always facing new obstacles and challenges that require innovative methods and systems to overcome (Akhtar and Rattani 2017).

2.2.4. Dimensionality Reduction

The Techniques that reduce how many input variables in a dataset are known as Dimensionality reduction, this makes tasks such as predictive modelling more challenging, this is known as the curse of Dimensionality. To Visualise the data, some techniques from Dimensionality reduction and some High-dimensionality statistics are used. The Use and combination of these techniques allow simplification of datasets, such as classification or regression datasets, to better fit a predictive model.

2.2.5. Evaluation Metrics

After training a predictive model, performance is determined by checking its accuracy, recall, precision and f1 score. It's the final process of facial recognition which helps in understanding the model. These values lie between 0 and 1. More the value better the performance, these values can be improved by tuning the model or changing the parameters to show difference in metrics, the final product will either increase the performance or decrease it (Brownlee 2020).

2.2.5.1. Accuracy

Evaluates a number between 0 and 1 based on the number of predictions correct in testing dataset. More the number better the accuracy. It determines the overall efficiency of the algorithm.

2.2.5.2. Precision

Precision checks the correctness of the result; it checks if the result is correctly identified by machine learning model.

2.2.5.3. Recall

Calculating the error rate present in the model, it's defined as number of positive results produced by the model. Higher the number means lower the error rate.

2.2.5.4. F1 Score

The mean between recall and precision is f1 score. A good f1 score is defined by having higher value between 0 and 1.

$$f1 = \frac{precision * recall}{precision + recall}$$

2.2.6. Different Face Recognition Methods

Throughout the years every work done on face recognition has been identified as a gateway to pursue and explore face recognition system in depth (Ghahramani 2015). Different face recognition methods used by researchers are:

- Template matching, Artificial Neural Network Approach.
- Local, Holistic and Hybrid- Classical Approach.
- Geometry and template, Face Descriptor Approach.

2.2.6.1. Comparison of Fisherface and Eigenface

Every image may not focus on facial features due to constant external disturbance, for example, in surveillance camera often it is quite difficult to identify a particular person if they are in motion. This hurdle can be crossed by applying appropriate method to develop the model. Distribution Based Methods have been proved to be quite resourceful while capturing a face image with expression. Comparatively, it is quite efficient and faster than eigenface based methods. In a recent article, a comparative study on performance was made between Eigenface and Fisherface by adding physical constraint to the dataset i.e., to check if the model was able to recognise the face in different weather conditions (Ahsan *et al.* 2021). The study showed that fisherface was able to provide better accuracy in different weather conditions especially during foggy and cloudy where most of the images are considered to be blurred.

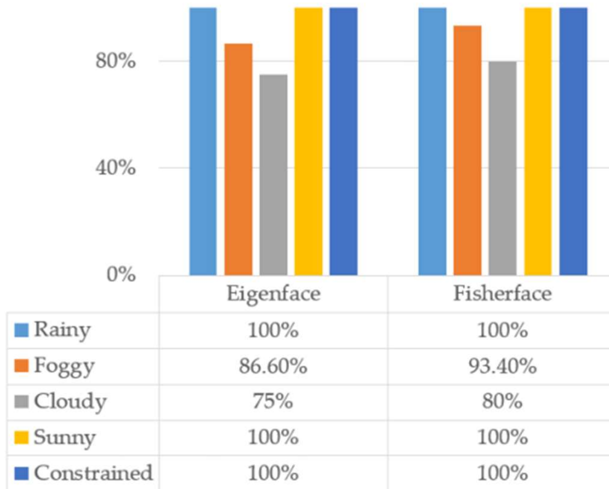

Figure 5 Performance Comparison between Eigenface and Fisherface (Ahsan et al. *2021)*

2.3. Related Work

There are lots of research published by the researchers on facial recognition, different methods and techniques were proposed to increase the precision and accuracy of the system, recent works have been proved to be quite efficient than older models by considering more factors while detecting a face. This section will discuss in detail of some related work done by researchers.

2.3.1. Artificial Neural Network Approach

As mentioned earlier, there are several factors to consider in face, while detecting a face, lots of information is required to find and identify the correct face print from the database. In a radical environment it is difficult to accumulate the required information especially during surveillance, exceptional performance is expected from the system. By using partial face print it is possible to negate this constraint. In a paper published by Reshma MR and Kannan B three major approaches were highlighted as optimal solution that supports partial face print as input (Reshma and Kannan 2019).

Before CNN, image processing was not possible in neural networks. After the development of CNN, it was known to perform well with images, and it follows the similar procedure of traditional neural network while computing images. The process of CNN is separated into three modules. Firstly, convolution, it selects the characteristics in an image, by identifying the characteristics the spatial relationship between pixels is preserved. A feature map is produced as an output which is in the form of matrix. Secondly, Pooling basically means sampling the feature map. Samples the matrix to reduce the length but without losing any important information about the image. Finally, the last step is flattening the image, the matrix is converted into linear sequence of array (Reshma and Kannan 2019). Using CNN as a baseline a new approach called Dynamic Feature Matching (DFM) is used for prediction of partial face. No additional pre-processing of images was required while input can be of any partial face with random size. Using fully convolutional neural network (FCN) for spatial feature, DFM can take advantage of FCN properties and produce accurate features. In the end, it is observed DFM approach shows positive impact on Face recognition and video recognition systems.

Another paper showed that CNN can be improved from base model by tuning. By applying improvised CNN on current face recognition algorithms, it can improve the precision and accuracy (Wang *et al.* 2020). Initially, changing the number of convolutional layers can impact the system, for comparative purposes parameters like number of samples and algorithms used were unchanged. The base model consists of eight layers, so it was altered to eleven by adding three more additional layers and incrementing the pooling layer by one. The number of filters used is around 128 with 3x3 matrix. The selected pooling window is 2x2 where the strides for both filters and pooling are 1. Secondly, three activation function was selected for image recognition, they are as follows:

- Tanh function
- RelU function
- elU function

Activation function is a trigger node that is placed between two neural networks, it decides if the neuron should be activated or not (Wang *et al.* 2020). The next change to improve the CNN is the correct usage of dropout functions. To negate the decline in accuracy, dropout functions are used to avoid overfitting while the model is in training phase. Using random selection, it deactivates certain amount of neurons. Improved accuracy of 85.39% was observed when the dropout rate is 100% while training. Finally changing the optimization algorithm that supports in increase in accuracy. Three different algorithms were studied while training the model:

- SGD
- RMprop
- Adam

It is observed Adam optimizer proved to better than the other algorithm with an accuracy of 88.41%, the reason is because Adam is an improved version of RMprop and dynamically changes the computation rate. Finally, the improved model showed an accuracy of 79.41% compared to base model with an accuracy of 68.85%.

Another important factor to consider while developing face recognition system is accurate prediction of gender from an image. The topic is proven to be delicate and if predicted inaccurately may violate the person's identity. To avoid such dire consequences, many paper was published to identify gender from an image. A researcher published a paper that can classify gender, the paper's primary objective was to overcome the error faced in image processing where the face and gender are not correctly recognised or synced in the process due to complex background resulting in inaccurate judgement of gender (Zhou *et al.* 2019). The given solution is to use CNN which is widely because of multiple image processing methods but mostly design to process pixel data. In CNN the images is processed in two different modules, face recognition and gender recognition. The first module detects the face in the image regardless of its gender and the second module characterizes the gender of the image, these modules are pretrained with data from LFW (Labelled face in wild), YTF (YouTube Face) which are basically public data's and they use to store the new face and recognise. Using AdaBoost, face is detected considering complex background, each image is resized and pixelated to 224 x 224. The normalized image is processed for alignment and prepared for face representation to identify the gender. Two different representations are used to build the model:

- Statistical Learning
- Knowledge Representation Method

ResNet 50 network architecture is used to construct the neural network with 50 layers. Using Global Average Pool (GAP) the size of the network is reduced because of the number of layers. The accuracy that gained by using this method is 1.42% (93.22%) of increment from that actual method which is 91.8%. The rest of the 6.88% of error occurs due to angle, illumination, and occlusion but this problem can be solved by multi-angled pictures of the same person (Zhou *et al.* 2019).

Computer vision is one of the modern fields in AI that helps in image processing. It is created by combining deep learning and CNN. It allows the system to study the image in higher level and produce meaning full information from digital images. A recommendation feature is implemented in computer vision where the user is prompted to take action (IBM solutions, 2020). One of the supported libraries from computer vision is OpenCV, it an image processing library focused on detecting images. A paper is published by combining computer vision technology and holistic based method in FR system to determine the performance. Appearance based FR system uses eigenface algorithm to detect and recognise face, but the performance is affected if the image focuses on illumination. Fisherface and LBP has the properties to overcome this problem and many researchers published a comprehensive analysis of LBP and Fisherface. This paper is based on comparison of face recognition in real time using Fisher's Linear Discriminant (FLD) with Local Binary Pattern (LBP), by using this method the paper proves that Fisherface has a good prediction and less time consuming than other methods like hybrid class and featured based. The author has implemented Fisher's linear discriminant to improve the classification rate to its peak value from within class to between-class (Jagtap *et al.* 2019). Along with this method the author used binary patterns for better recognition of face and managed to implement these methods with four basic stages of face recognition like face detection, pre-processing, featured extraction and matching the registered face. In face detection, Haar-cascade classifier was used to train all the faces and VISio dataset was chosen for training. During pre-processing, training was completed by using pre-registered images from the VISio dataset. In feature extraction phase, Fisherfaces and LBP algorithms were implemented using OpenCV library and for the matching the registered face, K-fold using cross validation was chosen. The above configuration achieved higher accuracy and significant increase in performance was observed (Jagtap *et al.* 2019).

2.3.2. Classical Facial Recognition Approach

Many research has been done using traditional approaches, due to rapid development in face recognition algorithms researchers have switched to modern approaches like CNN. Over the decade many papers have been published based on classical methods to prove improvements can be made when coupled with modern approaches. Classical approaches can be categorized into two class:

- Holistic Feature
- Local Feature Approach

In holistic approaches there are two well-known algorithms used in face recognition, Eigenface and Fisherface. In 2019, a paper was published by comparing the performance of two algorithms. The result showed that fisherface algorithm was better in terms of accuracy and time efficiency. When two algorithms were placed in simulation for detecting or recognizing the face the computational speed of fisherface was significantly faster than the other (Wahyuningsih *et al.* 2019). Each algorithm has their own distinct approaches in detecting images. Eigenface processes the image by calculating principal component analysis (PCA) to find the best vectors for the image, which is also called eigen vectors. PCA is a dimensionality reduction where it reduces the number of axes by discarding irrelevant information about the image. Using eigenvectors, eigenvalue is determined, this process is required to find the principal component number. Finally, using the number a covariance matrix is generated. Meanwhile, fisherface has three different sections involved in detecting the face, PCA calculation similar to eigenface, secondly, calculating fisher linear discriminant (FLD) and then compares to give more accurate result. Comparison is done through all these processes and the results are given as a graphical out-put, at the end of comparison, Fisherface Algorithm is 1% more accurate than Eigenface Algorithm and faster (Wahyuningsih *et al.* 2019).

As discussed, FR has created a huge impact on healthcare sector, FR system has passed many hurdles but can the system recognise if an individual's face is altered? A recent study focuses on predicting the correct face given the facial features is surgically altered. Using PCA for dimensionality reduction resolves the recognizing issues in 2D face images. The modified PCA is coupled with another algorithm called Viola-Jones, which is used to detect the face (George *et al.* 2017). The author explains by cropping the image to focus the face and resized to 64x64 pixels. By passing the input to the modified PCA, eigen value and vectors are computed. Passing the processed image to viola-jones algorithm converts the image to integral image. In the midst of the process, four key concepts are involved: Har like feature, Integral image, Ada Boost machine learning and Cascade classifier. The first attribute focuses on variations of black and white images. Integral image outlines the pixel values and uses for quick feature detection. Ada Boost focuses on the important characteristics of the image and ignoring the irrelevant information for example, background of the image is negated. Finally, the last attribute creates an enclosure where the face is present in the image, tries to detect the face (George *et al.* 2017). After training, when a new image is given as input to recognise, viola-jones algorithm detects the face in the image and if it results in true the image is cropped to focus on the required characteristics, using PCA similarity value is calculated. A pre-defined similarity value is given as default if the input's value is lesser than the default threshold value it will consider as a new face and store it in database. If the value is equal or higher than the threshold value, it will recognise the face even if the facial features are altered. The author stated the model can be enhanced to support noisy images and varied illuminated images.

An organisation where access is granted by face recognition prevents trespassing to unauthorized areas. But if the amount of device is less for a huge organisation, this may consume time to provide access for each employee. If the system can detect and recognise multiple faces, the process would be faster concurrently maintaining the accuracy of the model. A research paper that focuses on multiple face recognition using traditional approach was published to overcome the lengthy process (Mantoro *et al.* 2018a). The author used hybrid technique to achieve this goal, combining haar cascade and eigenface methods it was possible to detect up to 55 faces in a single detection. This improved model was able to display an accuracy of 91.67% (Mantoro *et al.* 2018b). Haar Cascade is implemented by using viola-jones algorithm, it is used to detect face in an image and PCA for dimensionality reduction.

Recognizing a face using partial image helped in reducing the processing speed but without loss of detailed feature extraction high computational speed cannot be achieved, to minimise the problem a paper proposes a solution by normalising expression and pose of a facial image. While detecting a face in real-time, it is not possible to get an expressionless face and the direction of the face may be captured in different axis. In this paper, the author describes the difficulties of FR when the facial image contains pose and expression, often leading in negative results. Proposed solution allows the face image to be converted into a single face neural reference (Petpairote *et al.* 2017). This step enables the system to create a synthesized face image that closely resembles or matches the face pattern present in the database. Using triangle transformation, two changes are made to the face image with expression, open-eyed and close mouth. By using the same transformation, the pose image is converted to frontal images. A face can exhibit different expression, rage, fear, smile, excitement, disgusted and so on. The author creates a virtual neutral face database, all the frontal neutral faces from the database are collectively wrapped to neutral face reference. During evaluation process, the virtual database is used for recognition purposes combined with synthesized frontal images. PCA and Local Binary Pattern (LBP) algorithms are used for facial recognition. Using piecewise affine wrapping (PAW) the expression images are referred with the virtual neutral face references to create synthesized images. After normalising the image LBP and PCA are applied. The proposed method showed a slight improvement in accuracy when compared with older models, an average accuracy of 95% was achieved using the wrapping method (Petpairote *et al.* 2017).

Another paper also provides a solution for different face position captured during real time detection. The author proceeds to analyse the problem by using PCA and LDA. While training the model, PCA is directed to an image and to determine the best fit classifier LDA is used. The author describes the problem by considering two factors, one is the location of the face and other is position in a cluttered background (Zhao *et al.* 1998). Without using CNN or other advanced approach, the paper focuses on getting accurate results by combining PCA and LDA. FERET dataset is used to train the model where PCA is used to construct a task specific space and LDA to perform classification. The hybrid approach proves to be useful by providing essential framework for other FR tasks. Before producing the results, assuming the images are normalised and sensitivity test was conducted by taking one image from the dataset and pre-processing the image. Three different properties were applied during the test, adding an artificial background, applying gaussian blur and pixel randomisation (Zhao *et al.* 1998).

In classical approach, many authors have proposed different solutions using PCA and Linear Discriminant Analysis but in the paper the author uses a different technique called Laplacianfaces to implement face recognition. It uses Locality Preserving Projections (LPP), the observed experimental result shows that the LLP finds the embedded components that preserves local information and obtains the face subspace meanwhile PCA and LDA are only found to be working on Euclidean structures of face space but not on all structures (He *et al.* 2005). The paper concludes that the Laplacianfaces is better in representation and gives a low error rate during the experiment while comparing to Eigenface and Fisher face method. The LLP method works by obtaining the optimal linear approximations to eigenfunctions of the LaPlace Betrami operator on the face manifold. By default, this method is identical to the linear method, however it has been discovered that by conserving the local structure, it can recover crucial parts of the intrinsic nonlinear main fold structure. In conclusion, LLP has more discriminating power than other methods like PCA and less sensitive to outliners. The error rate which was compared with Eigenfaces method and Laplacianfaces was that Laplacian faces had 14% less error compared to Eigenfaces also the Laplacianfaces method was compared with Fisher faces and resulted in 9% less error compared to FF method. The dims where less compared to Eigenfaces method and more than fisher faces method (He *et al.* 2005).

2.3.3. Face Descriptor-based Method

In this method there are different approaches to extract facial features. The local based method tries to produce global description but before producing the final component the local features of the face image which can be evaluated with neighbouring pixels present in the image (‘Konard Gromaszek’ 2021). In contrast to global approaches, which uses the complete image to generate each feature, pixel-level methods start with a pixel-level description of the face using the local neighbourhood of each pixel. The next phase involves diving the image into different subregions, for each subregion a local description is created. It is represented as a histogram of the pixel level description which was calculated in the initial stage. The information from the areas is then merged into the final descriptor by concatenating the partial histograms.

A study addresses facial recognition technology, which is the most difficult aspect of computer vision and pattern recognition. It contributes to the effectiveness of facial recognition programs. Facial recognition is highly vital in identifying an authorized individual, and it becomes much more critical when there are identical faces, twins, or the same human with varying ages. Initially, the Gabor filter technique was used to distinguish identical twins (Prema and Shanmugapriya 2017).The Gabor filter is a convolutional filter that combines gaussian and sinusoidal parameters. The weights are provided by the gaussian component, while the directionality is provided by the sine component. Gabor Kernel is a texture and edge recognition system that mimics the visual cortex. Many studies believe that neurons in the primary visual cortex of animals have orientated receptive field profiles with distinct spatial frequencies. Gabor filters may take use of important visual features including spatial localization, orientation selectivity, and spatial frequency. The other approach utilized in this study to distinguish identical twins using facial aspects is the multi-scale rapid radial symmetry transform. Gaussian pyramid construction, identification of primary facial features, fast radial symmetry detector, and Bipartite graph matching are all part of this procedure. Both techniques are put to the test, and the study finds that the results achieved by utilizing the multi-scale fast radial symmetry transform outperform the results obtained by using the Gabor filter method (Prema and Shanmugapriya 2017).

In the recent evaluation of biometrics system, the possibility of fraudulent access to system is simpler and more fragile. In this case, the method in the paper has provided a solution to deny and sturdy the total biometrics system by a novel software based fake detection method that can be used in multi-biometric system at same time to detect fragile activities that tries to gain access using a fake biometrics (Galbally *et al.* 2014) .The main grounds of the experiment is that to enhance the security of the biometric framework by including more liveness to the system and also maintaining the same faster and user friendly approach. The method represents a simple low degree of complexity so that it is easily enhanced according to the security level. The current biometric system is enhanced by using 25 general image quality features extracted from one image to highlight the difference between legitimate and spoofer samples. The system contains of sensors which is HW-Based liveness detection which detects the biometrics and then it is passed to feature Extractor where this system is also based on liveness detection and then it forwarded to the rest of the biometric system so when the system is tried to get spoofed it detects on first two stages of the entire system and alerts the environment and locks the permission to access any further. This experiment has concluded that the vulnerabilities of biometric systems against different type of attacks can be obstructed by the simple image processing method this method can be enhanced on the future with development of image processing. This method of developing the biometric has led to reduce the variability and updates the security of once biometric (Galbally *et al.* 2014).

Another paper published on face recognition using face descriptor method, shows feature extraction by implementing Fourier Descriptors (FD). Traditionally, the image is analysed using Fourier analysis in order to extract the shape. But in the recent years, there's been a significant growth in image processing and image analysis due to development of new technologies. To overcome the gap of development on image analysis this paper has proposed an enhanced Fourier Descriptor method (Wafi *et al.* 2016). The FD shape descriptor method is applied on images with different attributes like size, rotation and resolution. It is then used to compare with Fourier descriptors. Intra-class analysis is used to assess the performance of this technique, which is based on the total percentage of minimum absolute error with image rotation and scale translation. The results can be enhanced to produce more accurate values. The technique is then examined by a seven-step process initially taking original image, the next step is to rotate the image at an angle of 15°. The third step is to rotate the image again at an angle of 25°. Next the image is resized by 50% of its original image size. After applying all the methods, the implication factor of 1.5x is applied to the real size and then it is rotated to 25°. Finally, it is enlarged to 1.5X and saved by this the different angle and size of the image are being registered. By this enhancement of Fourier analysis, the new technique is more optimized and faster than the old Fourier descriptors based on the intra-class analysis (Wafi *et al.* 2016).

The evolution of image processing, image analysis and image recognition are considered to play an important role in modern day's security applications. In the past, methods based on local feature were a difficult task as it was necessary to build a robust Local Descriptor (LD). In this paper, the author considers the following factors important for a robust LD, remaining constant to variants in viewpoints, illumination, scale and rotation. The research tries to rationalise the robustness by implementing Scale Invariant Feature Transform (SIFT) descriptor (Zhou *et al.* 2014). The modern method is better than the old method which uses Sampling based Local Descriptor (SLD). The descriptor used in this method contains two properties; one it remains constants to the variations and second to search two controllable parameters. In the end, the image is controlled to find the optimal value by using these two properties for elliptical sampling. The test result was computed by using standard benchmark to highlight the robustness when the image is under constant variations.

2.4. Chapter Summary

This chapter summarised the different approach and technique proposed by researchers to tackle the problems in facial recognition. It details the purpose of PCA and Fisherface algorithm and different issues caused because of image variations. CNN was widely chosen by every researcher because of its popularity in processing images. This research will explore and highlight the performance report of PCA and Fisherface when coupled up with CNN.

3. Design And Implementation

3.1. Introduction

This chapter explains the procedure and algorithms used to implement Face Recognition. In the previous chapter different researchers published papers on Face Recognition by using different approaches and algorithms. From face detection to facial feature extraction, each step can be implemented by using different methods. Using CNN, it is proved that face recognition achieves better accuracy than other baseline methods. This dissertation tests the accuracy of CNN by coupling up with two holistic based algorithms to classify face data. Both the algorithms are compared with each other to show which one performs better. Minimum requirement for computation, steps involved in data collection, pre-processing and validation have been discussed in the subsequent sections of this chapter.

3.2. Software Requirement

3.2.1. Programming Language

The programming language used to create face recognition model is Python. Due to its simplicity, it is easy to learn and use. There are multiple programming languages but python is almost close to human level language while learning. Due to its simplified syntax, user base count is higher than other languages. The language is more than 30 years old with an active and supporting community. Due to the interactive community the language matured over the years and adequately assisted many developers. Renowned companies sponsored the language; hence it is highly recommended in many companies and schools. It contains hundreds of libraries and it also includes cloud media services that allows the user to work in different platforms (cross platforms). Supports any kind of virtual environment without performance loss. Current trending fields like cloud computing, data analyst and data science support python as their programming language, it is considered to be the second preferred language after R language.

Four factors that made python popular:

- Versatile in nature
- Efficient
- Speed
- Reliability

3.2.2. Python Libraries

3.2.2.1. Pandas

Pandas is an open-source python library that allows the programmers to manipulate the numerical data for analytical purposes. Before training a model, the dataset should be pre-processed like removing NULL values and assigning feature and label column for training. For data analyst and data scientist, pandas are used to simplify their work with time series.

3.2.2.2. NumPy

NumPy is another python package widely used after pandas, it supports mathematical functions to solve large multi-dimensional datasets. It helps in solving complex problems in machine learning.

3.2.2.3. Keras

A python library that focuses on deep learning. It's a high-level API used for implementing neural networks. Comparing with other frameworks, keras is slower because of high level abstraction in the frontend while having multiple options of backend computation, due to this it is highly recommended for beginners. It contains algorithms for optimisation, normalisation and activation layers. Different frameworks supported by keras are tensor flow, Theano and MXnet.

3.2.2.4. Matplotlib

Matplotlib is visual library that allows the programmers to plot arrays. It is built on NumPy arrays. Visually representing a dataset can create a better understanding. It is easier to understand patterns and determine correlations, used as a base tool required for reasoning about quantitative information.

3.2.2.5. Sklearn

It is a machine learning library that consists supervised and unsupervised algorithms. Dimensionality reduction, regression and classification are all included within the library for statistical modelling. Each process in machine learning requires sklearn to perform feature extraction and model tuning. PCA and Fisherface also requires sklearn library to use them.

3.2.2.6. OS module

Python allows the programmers to interact with the machine's operating system, by using OS library. File handling is done by importing OS library. It is a standard module that is installed along with python. Different functionalities can be accessed portably using OS. Renaming, creating, searching and locating directories (including files) is the primary focus of OS.

3.2.2.7. Jupyter Notebook

It's an open-source IDE to write and work with python code. Data analysts use this application for data manipulation and data processing. Different blocks of programming code can be used in the same notebook as it is differentiated in groups. Due to its flexibility, it is widely used by professional programmers.

3.3. Hardware Requirements

The configuration used for data processing is Intel(R) Core™ i7-10875H CPU with base clock frequency of 2.30GHz, 3000MHz of 16 GB Memory RAM and 1TB of SSD as hard disk. The model is built using Jupyter Notebook, the above configuration is sufficient to run computation multiple times for training and testing purposes.

3.4. Data Collection

There are wide range of datasets available for face recognition, the thesis aims to compare two different algorithms and evaluate the performance using neural networks. Hence, for the dissertation, dataset is taken from Labelled Face in the Wild (LFW), it consists of face images of up to 13,000 of same pixel size. Each image is labelled with respective name present in the picture and in the format of JPEG. UMASS created LFW library, it is a public benchmark for face verification (Huang *et al.* n.d.). Not all face images are used for face verification, as it takes up a lot of time to convert every image to .CSV format. Random celebrity folders are selected with considerate amount of images of same face. During the training phase the unlabelled images are passed for verification (Kawulok *et al.* 2016).

3.5. Data Pre-processing

The dataset consists of more than 13,000 facial images of celebrities and organized in alphabetical order. Initially the dataset is in JPEG format, so it must be converted into Comma Separated Value (CSV) format. The image will be presented in a spreadsheet with their pixel values. The size of the image is 250x250 hence the total number of columns should be 62,499 starting from 0. For this research, only few folders are selected randomly to convert them into training and testing set as it requires a lot of space to convert all the images. Initially, the folder's path is given using OS library. All the files under the folder and sub-folders are read. After reading all the images, it is converted into black and white (B&W) to reduce the size and to allow faster computation. Next, each image is now in the form of array, flattening of each image is done to change the multi-dimensional array to 1D array. This process further reduces the size of the image and takes less memory before passing it into the model. Each image is now in 1D array and appending them with the label. The label is a variable that holds the image's celebrity name. Finally, it is converted into CSV format for training and testing.

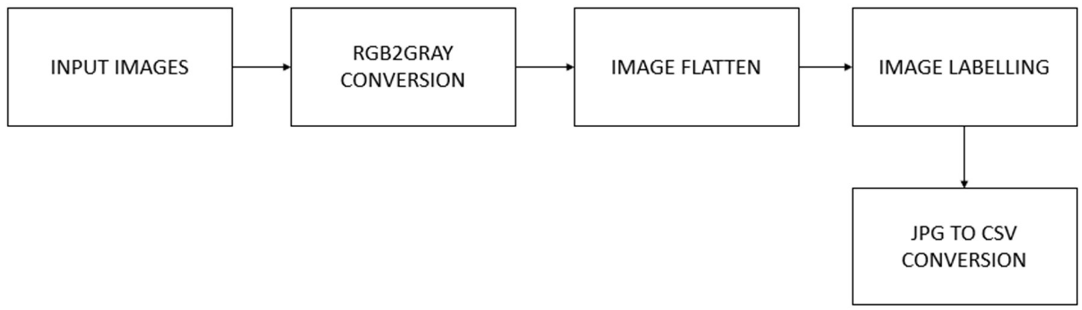

Figure 6 Image to CSV conversion

3.6. Exploratory Analysis

The dataset is already converted into CSV, each column represents each pixel of the image. A label column is appended to the dataset, which displays the celebrity's name of the image. The dataset doesn't consist of any duplicate or NULL values. Before splitting the dataset into training and testing, feature and label column is chosen to pass it as a parameter while splitting. Two variables are created, the feature column consists of all the pixel values of every image and the label variable takes the label column. Strings datatype cannot be passed into the label, so it is encoded using Label Encoder.

3.7. Feature Extraction

The feature column consists of all pixel value and it is used to generate feature vectors, two different algorithms are used to achieve this step. Principal Component Analysis (PCA) and Fisherface. The extracted feature is passed into CNN model for training. A mathematical overview of the algorithm is explained below, the experimental research will be explained in Result and Analysis chapter.

3.7.1. Principal Component Analysis

It's a dimensionality reduction technique, when the image has multiple components, PCA reduces the components as much as possible without losing any important information. It reduces the image to one dimension. It's an old algorithm widely used for face recognition or face detection. It converts the data into a simpler form in exchange for little accuracy, using orthogonal transformation the dimension of the image is reduced as much as possible. There are different steps involved in PCA ('Zakaria Jadi' 2021):

- Standardisation of value
- Form Covariance matrix
- Determine Eigen values and eigen vector using covariance matrix
- Find the number of principal components using eigen vectors and eigen value
- Feature vector

There are many initial variables that may differ in range, for example, one variable may lie between (1-100) and other may be around (5-10), the difference in the first variable is higher than the second. While analysing it will lead to biased result because initial variables with higher difference contribute more. To avoid this, standardisation is done, so every initial variable can contribute equally.

$$x = \frac{value - mean}{standard\ deviation}$$

Where x is the standardisation.

Forming a covariance matrix using the standardised data values is done. This is to determine the relationship between different variables, it is possible they contain information that may correlate to each other.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Using the covariance matrix, eigen values and eigen vectors are computed, with these two values numbers of principal components is determined. The number of components depends on the number of dimensions present in the data. Increment in number of principal components reduces the size of the image gradually by squeezing all information as much as possible.

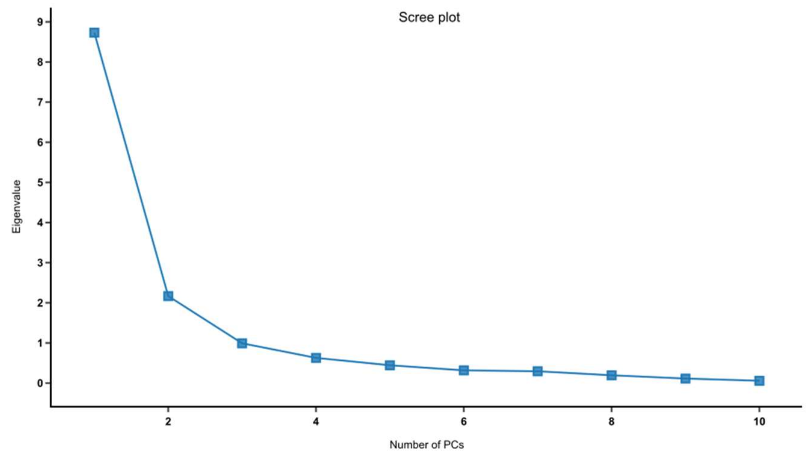

Figure 7 Variations captured after each principal component (' BioTuring's Blog' 2021)

3.7.2. Fisherface

A machine saves a face by differentiating variations in facial features (nose, eyes, face shape of the face). All the attributes are extracted as a feature by using eigenface. Fisherface uses same approach as eigenface, additionally computing the characteristic feature of the face vector. This research proposes to find the performance of Fisherface when used with CNN because it uses LDA for dimensionality reduction and has been proved by different researchers that LDA produces better results in terms of accuracy when compared with PCA (statement is supported by extensive research done in Chapter 2). The steps to implement fisherface are:

- Implement PCA for dimensionality reduction
- Apply Linear Discriminant (LDA) to obtain characteristic feature

Implement PCA to determine principal components, apply the components in LDA to obtain linear combinations of the image, LDA is also known as Fisher's Linear Discriminant. Unlike PCA, LDA is a supervised dimensionality reduction technique, LDA has two types of mathematical approach and it depends on number of classes. If the training set consists of two classes, two class LDA method is used if not multi-extension method is used. For this research, two class method is used to compute the results (Brownlee 2016).

Initially, two classes, C1 and C2 projections are observed to find any variations between them. Using the variation, d-dimensional mean vectors are calculated. μ is each point present in the class.

$$m_i = \begin{bmatrix} \mu_{\omega i\,(sepal\ length)} \\ \mu_{\omega i\,(sepal\ length)} \\ \mu_{\omega i\,(sepal\ length)} \\ \mu_{\omega i\,(sepal\ lengt\ \ )} \end{bmatrix}, \text{ with i = 1,2,3}$$

Secondly, scatter matrix is computed using the d-dimensional mean vectors. The use of scatter matrix is to maximise the separation of points in class C1 and C2.

$$S_w = \sum_{i=1}^{c} S_i$$

Where S_i is the scatter matrix for every class,

$$S_i = \sum_{i=1}^{c} (x - m_i)(x - m_i)^T$$

In the scatter matrix, eigen values and eigen vectors are computed, the vectors are sorted in the decreasing order of eigen values. A dimensional matrix (W) is formed by using N vectors with largest eigen values. The matrix W is used to convert the samples into new subset space.

$$A_\vartheta = \lambda_\vartheta$$

Where,

$\boldsymbol{A} = S_W^{-1} S_B$
$\boldsymbol{v} =$ Eigenvector
$\lambda =$ Eigenvalue

3.8. Building the Model

After feature extraction, the feature vectors are passed to CNN to train the model to produce results for testing dataset. CNN model is chosen because it is often used for image classification.

3.8.1. Convolutional Neural Network (CNN)

Before CNN, feature vectors were formed by using traditional algorithms like PCA. Since CNN allows image pre-processing like rotation and translation, it is easier to study image and extract content in higher detail. In this research, facial vectors are already formulated by using PCA and Fisherface but when coupled with CNN, the input is extensively processed to produce vector components in much higher standard. In general CNN consists of 3 layers, the first layer is called the convolutional layer, which does all the computation and it is also the core of CNN model. Throughout the process, a filter is applied through sliding window, so the image data can be convolved.

The second layer is the activation layer. Since images contain different objects that are not linear to each other, CNN uses a rectifier function to support the non-linearity.

The third layer is the pooling layer which down-samples the features. The pooling layer uses a non-overlapping filter to return the max value in the features. The final layer is the fully connected layer where the feature components are flattened. In the end, the output layer is classified by using an activation function. Figure 8 represents the different layer in CNN (Bansari 2019).

3.8.1.1. Hyperparameters in CNN

While defining a neural network, there are different hyperparameters declared to control the learning process. Through numerous iterations the hyperparameters can be tuned until the model fits.

3.8.1.1.1. Layers

A CNN model contains more than three layers, first is the input layer and the final layer is the output layer. Layers in between these two are considered as internal layer.

3.8.1.1.2. Activation Function

Creates non-linearity in the model. There are different activation functions like RelU, Sigmoid and tanh used until the model fits.

3.8.1.1.3. Batch Size

Number of samples given to CNN during training phase. Increasing the batch size will increase the computation process and memory consumption. The size could be 16,32,64 and so on.

3.8.1.1.4. Epochs

Number of times the training set is found in the neural network. The value increases until a small gap appears between test and training error.

3.8.1.1.5. Filters

To identify different patterns in the image dataset. The number of filters depends on the complexity of the image and its detail.

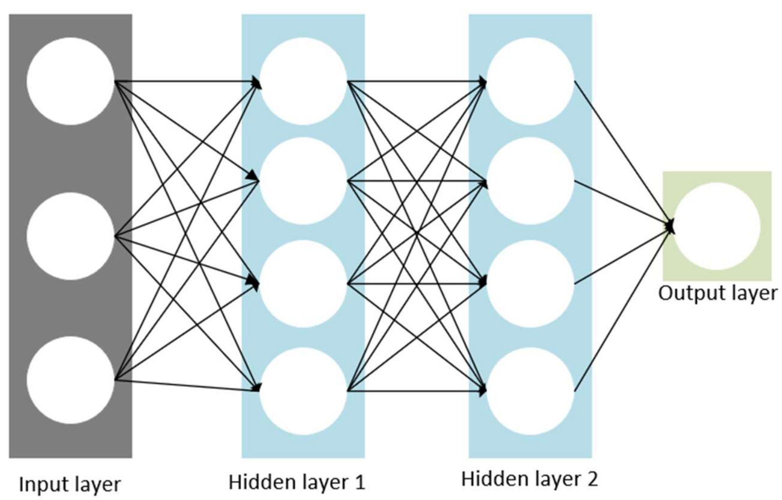

Figure 8 CNN Layer architecture

3.9. Implementation

3.9.1. System Architecture

The complete architecture of the model is represented below, from pre-processing to feature extraction and training the model using CNN is shown. Shortly summarized, Image is passed for pre-processing where it is converted into NumPy array and saved in CSV format, the label column in the data frame is encoded and the dataset is split for training and testing. Feature vector is formulated by applying PCA and FDA and finally passed to CNN for training, after training the testing dataset is passed to determine the accuracy.

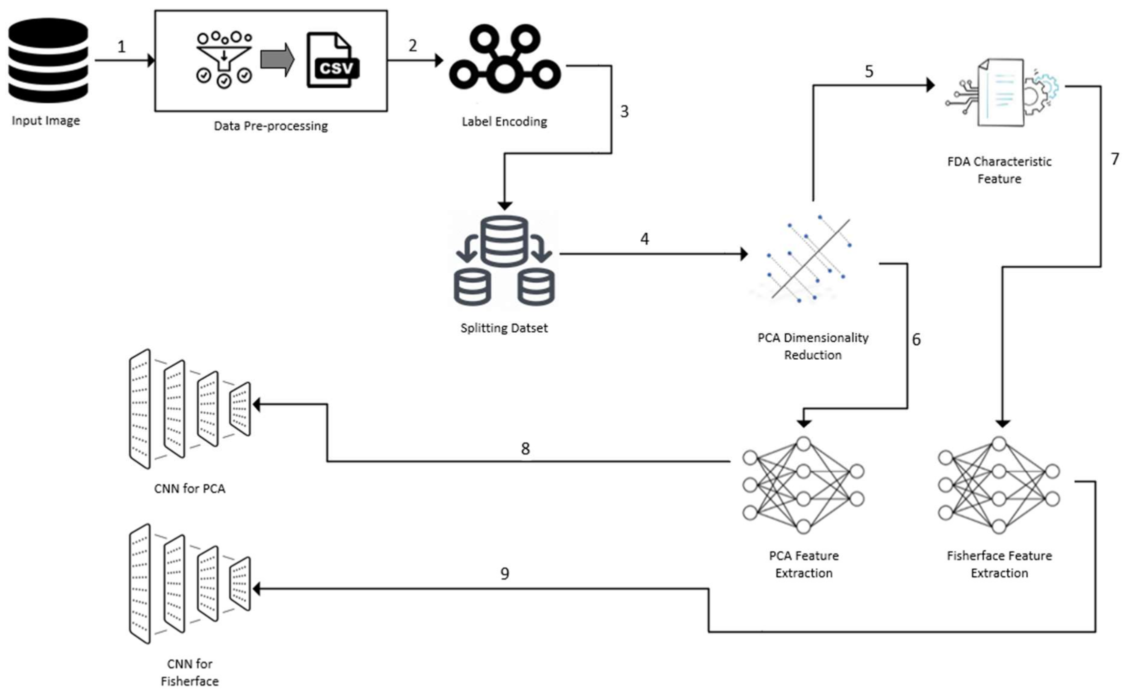

Figure 9 Face Recognition System Architecture

3.9.2. Data Pre-processing

The images are converted to a series of array, so that it can be understood by the algorithm and using pandas it is visually represented in a table. In this research, four folders of celebrity's images are chosen randomly for training, the number of different celebrities will be added later. And finally, a folder named "unknown" is created and two images of four celebrity are moved to the unknown folder for testing. Figure 10 shows sample images in the dataset before converting.

Figure 10 Sample Dataset

All the images are stored in local machine. Total images present in all 5 folders combined is 33.

```
D:/thesis_dataset/Abdoulaye_Wade\Abdoulaye_Wade_0001.jpg
D:/thesis_dataset/Abdoulaye_Wade\Abdoulaye_Wade_0002.jpg
D:/thesis_dataset/Abdoulaye_Wade\Abdoulaye_Wade_0003.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0001.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0002.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0003.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0004.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0005.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0006.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0007.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0009.jpg
D:/thesis_dataset/Adrien_Brody\Adrien_Brody_0011.jpg
D:/thesis_dataset/John_McCain\John_McCain_0001.jpg
D:/thesis_dataset/John_McCain\John_McCain_0002.jpg
D:/thesis_dataset/John_McCain\John_McCain_0005.jpg
D:/thesis_dataset/John_McCain\John_McCain_0006.jpg
D:/thesis_dataset/John_McCain\John_McCain_0007.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0001.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0003.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0004.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0005.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0006.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0007.jpg
D:/thesis_dataset/Paradorn_Srichaphan\Paradorn_Srichaphan_0008.jpg
D:/thesis_dataset/Unknown\Abdoulaye_Wade_0004.jpg
D:/thesis_dataset/Unknown\Adrien_Brody_0008.jpg
D:/thesis_dataset/Unknown\Adrien_Brody_0010.jpg
D:/thesis_dataset/Unknown\Adrien_Brody_0012.jpg
D:/thesis_dataset/Unknown\John_McCain_0003.jpg
D:/thesis_dataset/Unknown\John_McCain_0004.jpg
D:/thesis_dataset/Unknown\Paradorn_Srichaphan_0002.jpg
D:/thesis_dataset/Unknown\Paradorn_Srichaphan_0009.jpg
D:/thesis_dataset/Unknown\Paradorn_Srichaphan_0010.jpg
```

Figure 11 Conversion of Images

Figure 11 shows that all the images have been successfully converted. Next, Figure 12 displays the code to save dataset in the local machine

In [5]:
```
#Save the dataframe
df.to_csv('D:/thesis_dataset/ultimate_test.csv')
```

Figure 12 Saving the CSV file

Now to display the information in a table that has been converted. Figure 13 shows the pixel values of each image and with known label name and the Figure 14 shows the same information but without knowing the celebrity's name.

Out[3]:

	0	1	2	3	4	5	6	7	8	9	...	62491	62492	62493	62494	62495	62496	62497	62498	62499	label
0	214	214	213	213	213	214	215	215	218	218	...	53	59	58	54	6	2	1	1	1	Abdoulaye_Wade
1	45	47	48	48	49	50	51	51	52	55	...	34	34	34	34	34	0	1	0	0	Abdoulaye_Wade
2	1	1	1	1	1	0	0	0	0	0	...	45	1	0	0	1	0	0	0	1	Abdoulaye_Wade
3	0	0	0	0	0	0	0	0	7	7	...	50	53	53	52	52	1	0	0	1	Adrien_Brody
4	108	105	99	93	87	79	69	61	54	50	...	47	49	51	56	60	65	69	79	76	Adrien_Brody

5 rows × 62501 columns

Figure 13 Image data frame (Training)

Out[6]:

	0	1	2	3	4	5	6	7	8	9	...	62491	62492	62493	62494	62495	62496	62497	62498	62499	label
28	0	0	0	0	0	0	0	0	0	0	...	6	6	5	5	6	6	4	1	1	Unknown
29	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	Unknown
30	4	6	8	6	5	6	9	11	8	9	...	1	1	0	1	2	1	0	0	0	Unknown
31	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	Unknown
32	0	0	0	0	0	3	6	6	7	7	...	0	0	0	0	0	0	0	0	0	Unknown

5 rows × 62501 columns

Figure 14 Image data frame (Testing)

3.9.3. Exploratory Analysis

As discussed, feature is pixel columns (0 – 62,499) and label is the label column. Before splitting the dataset, the label is in the datatype of string which needs to be converted to object, so label encoding is done.

```
Out[8]: array(['Abdoulaye_Wade', 'Adrien_Brody', 'John_McCain',
               'Paradorn_Srichaphan', 'Unknown'], dtype=object)
```

Figure 15 Label Encoding

Figure 15 shows the different labels in the label column and Figure 16 shows the encoded form of each image.

```
[0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4]
```

Figure 16 Encoded Values

Each encoded value is appended with a celebrity's name and represented in Table 1 for reference.

Table 1 Encoded Labels

Label	**Celebrity's Name**
0	Abdoulaye Wade
1	Adrien Brody
2	John McCain
3	Paradorn Srichaphan

After choosing the feature and label column, a function is defined to display the image in the feature column.

Figure 17 Displaying the images in the dataset using pixel values

In Figure 17 images are displayed by using the points in each column and it can be plotted using matplot library. Now to split the dataset in the ration of 0.8 for training and 0.2 for testing (Figure 18).

```
In [14]: x_train.shape
Out[14]: (24, 62500)

In [16]: x_test.shape
Out[16]: (9, 62500)
```

Figure 18 Splitting the dataset

3.9.4. Feature Extraction

3.9.4.1. PCA

Before extracting feature, the dataset is scaled using Standard Scalar because while model training the feature behaved badly. This happens if the feature is more or less standard in normal distributed data. Figure 19 shows the code to standardise the feature in training and testing set.

```
In [44]: #Standard Scaling
         scaler = StandardScaler()
         scaler.fit(x_train)
         X_sc_train = scaler.transform(x_train)
         X_sc_test = scaler.transform(x_test)
```

Figure 19 Standard Scaling done on training set

The next process is to extract feature using PCA.

```
(24, 62500)
(24, 15)
(24,)
[[1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 1. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 0.]]
```

Figure 20 Feature extracted using PCA

Figure 20 displays the feature vectors of training dataset, after continuous iteration the number of components chosen for this research is 15. It is 15 because by referring Figure 21 there isn’t much variation after 14 in the cumulative variance, which means the information is either not relevant or already similar to the stored information. Hence, it is neglected. This process reduces the size and dimensionality of the image with increase in number of components.

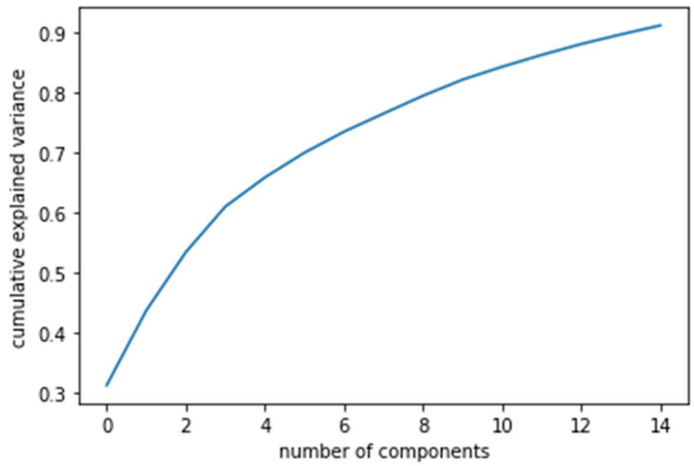

Figure 21 Cumulative variance for each component in PCA

The sampled images shown in Figure 22 is after extracting the feature, after each increment in number of components the images are blurry and after 15 the images are determined to be quite similar to previous components ('HEROKU' 2021).

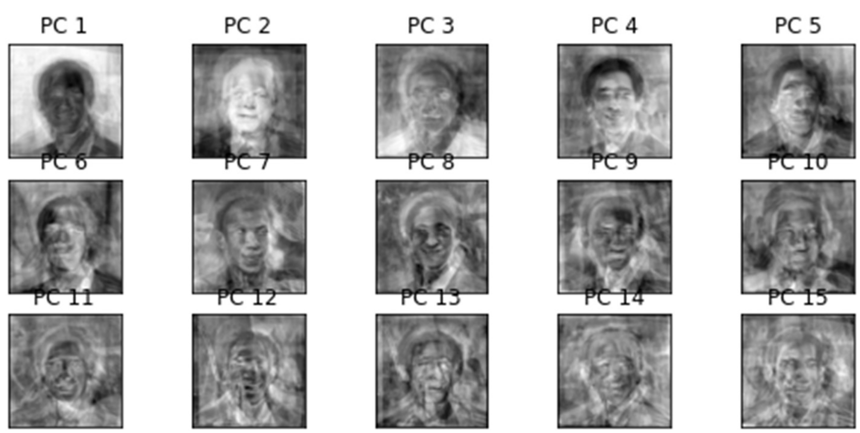

Figure 22 PCA feature vector

3.9.4.2. Fisherface

Using the feature vector formed by PCA is passed to LDA which is also called Fisher's Linear Discriminant (FLD). The vectors are transformed further by using LDA to determine the new subspace in the sample (Refer Figure 23).

```
(24, 62500)
(24, 3)
(24,)
[[1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 1. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 0.]]
```

Figure 23 Feature extracted using Fisherface

Using the Figure 24, the number of components is 3, as the images have already been reduced to its simplest form in PCA. In the graph, there is just a small variation observed when the component is chosen 1, so it is chosen 3 just to consider the new cumulative variance.

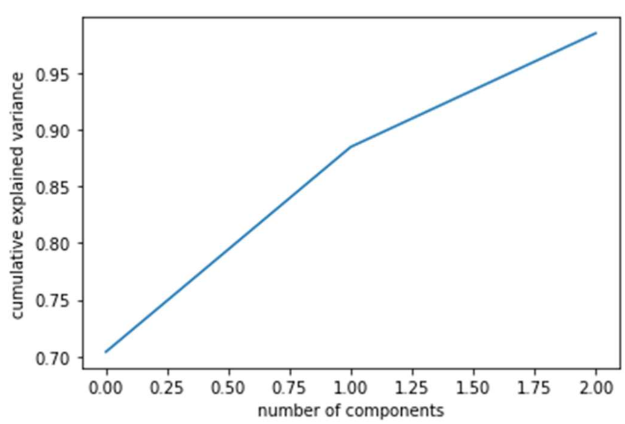

Figure 24 Cumulative variance for each component in Fisherface

3.9.5. Model Training

Model is trained using CNN from the feature extracted using PCA and Fisherface algorithms. For this research same parameters is passed for both the algorithms and results are evaluated.

3.9.5.1. Training for PCA using CNN

For this research, sequential model is chosen with one layer and 128 units, the units represent the number of neurons present in the layer. Defining a dense layer for input with the activation function “relu” along with a regularisation layer. One set of hidden layers is included in the internal layer, it consists of one dense layer and two regularisation layers. The hidden layer is used for training the dataset. Now output layer is defined for validation, it is done by adding dense layer with activation function “softmax”. Since the dataset has only 5 labels after extracting the feature so the total number label given in the output layer is 5. Defining a function to compile the model before training. During compilation, the optimiser chosen is “rmsprop” and loss function is “categorical_crossentropy” and metrics is “categorical accuracy”. These parameters were chosen carefully after multiple iterations.

For training the model, the hyperparameters are tuned and altered after each iteration to produce different results for comparison. The current standardised value chosen for each hyperparameters; epoch is 100, batch size is 256 and validation split is 0.15.

Table 2 Model training summary for different Hyperparameters (PCA)

Batch Size	Epoch	Categorical Accuracy	Categorical Validation
32	50	0.45	0.25
64	50	0.25	0.50
64	100	0.55	0.50
128	100	0.75	0.50
256	100	0.50	0.75

 Table 2 shows spike in categorical validation by increasing the batch size periodically and choosing number of instances as 100. The lowest categorical validation is observed when the batch size is 32 with 50 instances.

After training the model for PCA with CNN, the summary of the model is sohwn in Figure 25.

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 128)               2048
_________________________________________________________________
gaussian_noise_12 (GaussianN (None, 128)               0
_________________________________________________________________
dense_19 (Dense)             (None, 128)               16512
_________________________________________________________________
gaussian_noise_13 (GaussianN (None, 128)               0
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_20 (Dense)             (None, 5)                 645
=================================================================
Total params: 19,205
Trainable params: 19,205
Non-trainable params: 0
```

Figure 25 Model Summary using PCA

3.9.5.2. Training for Fisherface using CNN

For Fisherface, the feature is extracted from LDA, the input passed for LDA was previously extracted from PCA, so the set of samples subspaces have been reduced as much as possible. For training, the model chosen is Sequential and number of layers is 1. The activation function for input layer is same as PCA and realisation layers are also the same. The hidden layer also consists of one dense and two realisation (gaussian and dropout) layers. The compile function is also same as PCA. The hyper parameter values are standard and altered after each iteration.

Table 3 Model training summary for Different Hyperparameters (Fisherface)

Batch Size	**Epoch**	**Categorical Accuracy**	**Categorical Validation**
32	50	0.40	0.50
64	50	0.55	0.50
64	100	0.55	0.50
128	100	0.45	0.50
256	100	0.40	0.75

The categorical validation is consistent throughout the training until the batch size is 256 and running it 100 times (Refer Table 3). After training the model for Fisherface with CNN, the summary of the model is displayed in Figure 26.

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_21 (Dense)             (None, 128)               512
_________________________________________________________________
gaussian_noise_14 (GaussianN (None, 128)               0
_________________________________________________________________
dense_22 (Dense)             (None, 128)               16512
_________________________________________________________________
gaussian_noise_15 (GaussianN (None, 128)               0
_________________________________________________________________
dropout_7 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_23 (Dense)             (None, 5)                 645
=================================================================
Total params: 17,669
Trainable params: 17,669
Non-trainable params: 0
```

Figure 26 Model Summary using Fisherface

3.10 Chapter Summary

This chapter explains the system architecture of the model and each components used to build the model. The background process of feature extraction algorithms and CNN model is explained in detail. The next chapter discusses the experimental result obtained after running the model.

4. Result And Analysis

This chapter focuses on performance of two feature extraction algorithms using CNN as a baseline. Through multiple iterations, the results were obtained and after each iteration the parameters were tuned based on the number of images given as input. Only the best performance obtained from PCA and Fisherface algorithm using CNN is displayed for comparison.

4.1. Model Prediction for PCA using CNN

From figure 27, the following result displays the accuracy of the model given the batch size is 256 and number of instances is 100 with split ratio of 0.15. The model has been trained to identify four different celebrities given that the number images of each celebrity is more than 3. For this experiment, the testing set contains more than 2 images of four celebrities labelled as unknown. Using PCA as the primary feature vector, the average accuracy observed is 0.56 with f1 score of 0.50, for 9 samples. The f1 score is the average weight of precision and recall.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.50	1.00	0.67	2
2	1.00	0.50	0.67	2
3	0.50	1.00	0.67	1
4	0.50	0.33	0.40	3
accuracy			0.56	9
macro avg	0.50	0.57	0.48	9
weighted avg	0.56	0.56	0.50	9

Figure 27 Evaluation report for testing set

The first column in figure 27 is the encoded celebrities name, no faces were recognised for the label 0 and for the label 2, the model was able to predict all the faces present in the testing set. These values vary if more images are passed for training which allows the PCA to increase the number of components. The predicted result is saved by defining a function called “write_prediction”. Figure 28 shows the code to save the predicted result.

```python
def write_predictions(predictions1, fname):
    pd.DataFrame({"ImageId": list(range(1,len(predictions1)+1)), "Label": predictions1}).to_csv(fname, index=False, header=True)

write_predictions(predictions1, "pca-keras-mlp.csv")
```

Figure 28 Code to save the result

The saved csv file shows the face recognised by PCA for each sample.

```
ImageId,Label
1,Adrien_Brody
2,Adrien_Brody
3,Adrien_Brody
4,Unknown
5,John_McCain
6,Paradorn_Srichaphan
7,Unknown
8,Paradorn_Srichaphan
9,Adrien_Brody
```

Figure 29 Recognised faces using PCA

4.2. Model Prediction for Fisherface using CNN

Using the same hyperparameter values as PCA, the test dataset produced the following result with an average accuracy of 0.67 and f1 score of 0.61 for 9 samples.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	0.67	1.00	0.80	4
2	0.00	0.00	0.00	0
3	1.00	0.50	0.67	2
4	0.00	0.00	0.00	2
accuracy			0.67	9
macro avg	0.53	0.50	0.49	9
weighted avg	0.63	0.67	0.61	9

Figure 30 Evaluation report for Fisherface

Similarly defining a function to save the predicted result in a csv file. The saved csv file displays the face recognised by Fisherface for each sample (Refer Figure 31).

```
ImageId,Label
1,Paradorn_Srichaphan
2,Adrien_Brody
3,Adrien_Brody
4,Adrien_Brody
5,John_McCain
6,Adrien_Brody
7,Abdoulaye_Wade
8,Adrien_Brody
9,Adrien_Brody
```

Figure 31 Faces recognised using Fisherface

4.3. Performance Comparison between PCA and Fisherface

The result was computed by increasing the number of training images and increasing the number of samples of each celebrity. The comparison table displays the average accuracy of the model trained by Fisherface and PCA.

Table 4 Accuracy Comparison Between PCA and Fisherface (Testing)

Training Samples	PCA Average Accuracy	Fisherface Avg. Accuracy
24	0.56	0.67
40	0.47	0.55
50	0.50	0.60
60	0.35	0.47

By referring Table 4, the result shows that the accuracy obtained by Fisherface when trained with CNN model maintains are consistent accuracy between 0.50 and 0.60 in average. But for PCA the lowest accuracy obtained is 0.35 and the highest is 0.56, this clearly states that Fisherface performs better than PCA in all observed cases.

Table 5 Performance Comparison Between PCA and Fisherface (Testing)

Evaluation Metrics	Fisherface	PCA
Accuracy	0.67	0.56
Precision	0.63	0.56
Recall	0.67	0.56
F1 Score	0.61	0.50

Accuracy is one of the important factors to estimate the performance of the model. After extensive research done in Chapter 2, Fisherface does perform better than PCA. But there are evaluation metrics that affect the accuracy of model. Using Table 5, it is clear that Fisherface produced a better validation score than PCA by achieving high F1 score, the model behaves better with higher f1 score.

4.4 Chapter Summary

This chapter explains displays the output of the model and determines the performance of PCA and Fisherface when trained using CNN. Performance report for each algorithm is displayed in the form of table. The next chapter discusses about the conclusion and future work of the thesis.

5. Conclusion and Future Works

5.1. Conclusion

Over the years, there's been a significant improvement in Face Recognition technology, as it proves to be beneficial for different industries. Almost every organisation has invested their research and development to deploy this technology with primary focus on improving security. As shown in related work, researchers have been trying to build new algorithms or improve existing models to recognise faces more efficiently. In the recent years, working with image dataset has been simplified because of Convolutional Neural Network algorithms. This study shows the experimental result to recognise face using CNN as a baseline, which is coupled up with Principal Component Analysis and Fisherface algorithms.

Labelled Faces in the Wild (LFW) dataset was used perform this experiment, it's an open-source database that contains face photograph of different celebrities. Total of 33 images were used, it had training set of 24 images and 9 images of testing set. Throughout the process, same parameters were set for both the algorithms.

5.2. Performance of Fisherface using CNN

Before highlighting the performance of Fisherface using CNN, two statements are defined which is associated with the research.

CNN is a newly developed technique that works well with image classification and Fisherface is a traditional algorithm widely used in the 90's for extracting facial feature in an image. To avoid compatible errors, the CNN model is configured to accept the feature vector of Fisherface.

The CNN model is used as a baseline and not for feature extraction. The facial vectors are extracted using Fisherface.

The performance of the model is determined by evaluating the results. Accuracy, precision, recall and f1 score are the four metrics that measures the performance of the model. By referring figure 30, an accuracy of 67%, precision of 63%, recall of 67% and f1 score 61% was observed. It indicates that model was able to recognise more than 60% of faces in the testing dataset.

5.3. Performance Comparison Between PCA and Fisherface

CNN was used for training the model, the feature was extracted by PCA and Fisherface separately and each feature vector was passed to CNN to compare the performance between them. After evaluating the results, using table 4 and 5 it is found that CNN coupled with Fisherface performed better by achieving an accuracy of 67% while PCA produced an accuracy of 56%. By referring the performance table present in chapter 4, it is shown that Fisherface achieved more precision, recall and better f1 score than PCA.

5.4. Limitations

The research focused on coupling up a traditional face recognition algorithm with CNN. In this paper, to implement PCA and Fisherface, the image dataset was converted into CSV table, each column represented pixel value. Since the default size of the image is 250x250, each image in CSV consists of 62500 columns. Even though the image was converted to black and white the size of CSV file was high, hence small numbers of images were chosen for training and testing.

In this research, feature is extracted by using Fisherface and PCA. Even though both the algorithms are relatively simple to implement than CNN, they don't perform well if the image contains expression or variation in pose. PCA struggles in identifying facial expression or if the image contains complex background and Fisherface performance declines if there are variations in pose. To avoid these constraints, the dataset chosen for training and testing contains simple characteristic features.

5.5. Future Works

A machine learning model is an iterative model, given the time and resources there are different areas to focus on improving the model. Initial drawback of the model is the image conversion which occupies a lot of space in the memory, it can be avoided by streaming the image directly by using Computer Vision (CV). Using cv2 library the image attributes can be manipulated. As it is specifically designed to work with images there are different face recognition algorithms that can be imported. Resize, colour conversion and morphing of images can be done easily. By using cv2, large dataset can be used for training which helps in improving the accuracy of the model.

The model uses PCA and Fisherface for feature extraction and uses CNN as baseline. The performance can be improved if CNN is used to extract face components because CNN works well with complex background, can identify facial expression unlike PCA and the performance doesn't decline if there's any variation in pose. Also, by using different CNN model there can be improvement in performance.

Appendices

Appendix A

It contains the references to all the source of material which were accessed in the process of completing the thesis.

Appendix B

This section contains the code listing, list of python codes to implement the model.

Appendix A: References

A Step-by-Step Explanation of Principal Component Analysis (PCA) [online] (2021) *Built In*, available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis [accessed 30 Aug 2021].

Ahsan, M.M., Li, Y., Zhang, J., Ahad, M.T., Gupta, K.D. (2021) 'Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions', *Technologies*, 9(2), 31.

Akhtar, Z., Rattani, A. (2017) 'A Face in any Form: New Challenges and Opportunities for Face Recognition Technology', *Computer*, 50, 80–90.

Bansari, S. (2019) Introduction to How CNNs Work [online], *Medium*, available: https://medium.datadriveninvestor.com/introduction-to-how-cnns-work-77e0e4cde99b [accessed 31 Aug 2021].

Brownlee, J. (2016) 'Linear Discriminant Analysis for Machine Learning', *Machine Learning Mastery*, available: https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/ [accessed 23 Aug 2021].

Brownlee, J. (2020) 'Tour of Evaluation Metrics for Imbalanced Classification', *Machine Learning Mastery*, available: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/ [accessed 31 Aug 2021].

CODE HEROKU DASHBOARD [online] (2021) available: https://www.codeheroku.com/course_content?lesson_id=0&course_id=1&section_id=0 [accessed 31 Aug 2021].

crime, J.S.J.S. writes about, justice, News, related matters for N. (2021) 'We Try to Use It as Much as We Can': How Facial Recognition Became a Routine Policing Tool in America [online], *NBC News*, available: https://www.nbcnews.com/news/us-news/how-facial-recognition-became-routine-policing-tool-america-n1004251 [accessed 30 Aug 2021].

Face Recognition: Issues, Methods and Alternative Applications | IntechOpen [online] (2021) available: https://www.intechopen.com/chapters/51031 [accessed 30 Aug 2021].

Facial Recognition Cameras Will Put Us All in an Identity Parade | Frederike Kaltheuner [online] (2020) *the Guardian*, available:

http://www.theguardian.com/commentisfree/2020/jan/27/facial-recognition-cameras-technology-police [accessed 30 Aug 2021].

Facial Recognition Global Market Size 2025 [online] (2021) *Statista*, available: https://www.statista.com/statistics/1153970/worldwide-facial-recognition-revenue/ [accessed 30 Aug 2021].

Galbally, J., Marcel, S., Fierrez, J. (2014) 'Biometric Antispoofing Methods: A Survey in Face Recognition', *IEEE Access*, 2, 1530–1552.

George, G., Boben, R., Radhakrishnan, B., Suresh, L.P. (2017) 'Face recognition on surgically altered faces using principal component analysis', in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, Presented at the 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), 1–6.

Ghahramani, F. (2015) *Face Recognition: An Engineering Approach*, Master of Science, available: https://scholarworks.sjsu.edu/etd_theses/4635 [accessed 31 Aug 2021].

Global Contactless Biometrics Market Revenue 2027 [online] (2021) *Statista*, available: https://www.statista.com/statistics/1154004/worldwide-contactless-biometrics-market-revenue/ [accessed 30 Aug 2021].

He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.-J. (2005) 'Face recognition using Laplacianfaces', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), 328–340.

How to Read PCA Biplots and Scree Plots - BioTuring's Blog [online] (2021) available: https://blog.bioturing.com/2018/06/18/how-to-read-pca-biplots-and-scree-plots/ [accessed 30 Aug 2021].

Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E. (n.d.) 'Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments', 11.

Infographic: Americans Accept Facial Recognition for Public Safety [online] (2021) *Statista Infographics*, available: https://www.statista.com/chart/19321/facial-recognition-public-opinion/ [accessed 30 Aug 2021].

Jagtap, A.M., Kangale, V., Unune, K., Gosavi, P. (2019) 'A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV', in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, Presented at the 2019 International Conference on Intelligent Sustainable Systems (ICISS), 219–224.

Kawulok, M., Celebi, M.E., Smolka, B. (Eds.) (2016) *Advances in Face Detection and Facial Image Analysis* [online], Springer International Publishing: Cham, available: http://link.springer.com/10.1007/978-3-319-25958-1 [accessed 30 Aug 2021].

Lu, P., Song, B., Xu, L. (2021) 'Human face recognition based on convolutional neural network and augmented dataset', *Systems Science & Control Engineering*, 9(sup2), 29–37.

Mantoro, T., Ayu, M.A., Suhendi (2018a) 'Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods', in *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, Presented at the 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), 1–5.

Mantoro, T., Ayu, M.A., Suhendi (2018b) 'Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods', in *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, Presented at the 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), 1–5.

Ondrišová, M., Molnár, L., Juhás, G., Juhásová, A., Mažári, J., Mladoniczky, M. (2019) 'Story of Artificial Intelligence: research and publications behind', in *2019 17th International Conference on Emerging ELearning Technologies and Applications (ICETA)*, Presented at the 2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA), 581–586.

Petpairote, C., Madarasmi, S., Chamnongthai, K. (2017) 'A pose and expression face recognition method using transformation based on single face neutral reference', in *2017 Global Wireless Summit (GWS)*, Presented at the 2017 Global Wireless Summit (GWS), 123–126.

Prema, R., Shanmugapriya, P. (2017) 'A review: Face recognition techniques for differentiate similar faces and twin faces', in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Presented at the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2899–2902.

Reshma, M.R., Kannan, B. (2019) 'Approaches on Partial Face Recognition: A Literature Review', in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Presented at the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 538–544.

Sithara, R., Rajasree, R. (2019) 'A survey on Face Recognition Technique', in *2019 IEEE International Conference on Innovations in Communication, Computing and Instrumentation*

(ICCI), Presented at the 2019 IEEE International Conference on Innovations in Communication, Computing and Instrumentation (ICCI), 189–192.

Sumathi, S., RaniHema Malini, R. (2010) ‘Face recognition system to enhance E Health’, in *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, Presented at the 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), 195–198.

Wafi, N.M., Yaakob, S.N., Salim, N.S., Jusoh, M., Nazren, A.R.A., Hisham, M.B. (2016) ‘Image analysis using new Descriptors Average Feature Optimization based on Fourier Descriptors technique’, in *2016 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, Presented at the 2016 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), 135–138.

Wahyuningsih, D., Kirana, C., Sulaiman, R., Hamidah, Triwanto (2019) ‘Comparison Of The Performance Of Eigenface And Fisherface Algorithm In The Face Recognition Process’, in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Presented at the 2019 7th International Conference on Cyber and IT Service Management (CITSM), 1–5.

Wang, D., Yu, H., Wang, D., Li, G. (2020) ‘Face Recognition System Based on CNN’, in *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, Presented at the 2020 International Conference on Computer Information and Big Data Applications (CIBDA), 470–473.

What Is Facial Recognition? - Definition from WhatIs.Com [online] (2021) *SearchEnterpriseAI*, available: https://searchenterpriseai.techtarget.com/definition/facial-recognition [accessed 30 Aug 2021].

Yohanes, B.W., Diaz Airlangga, R., Setyawan, I. (2018) ‘Real Time Face Recognition Comparison Using Fisherfaces and Local Binary Pattern’, in *2018 4th International Conference on Science and Technology (ICST)*, Presented at the 2018 4th International Conference on Science and Technology (ICST), 1–5.

Zhao, W., Chellappa, R., Krishnaswamy, A. (1998) ‘Discriminant analysis of principal components for face recognition’, in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Presented at the Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, 336–341.

Zhou, W., Wang, C., Xiao, B., Zhang, Z. (2014) ‘SLD: A Novel Robust Descriptor for Image Matching’, *IEEE Signal Processing Letters*, 21(3), 339–342.

Zhou, Y., Ni, H., Ren, F., Kang, X. (2019) ‘Face and Gender Recognition System Based on Convolutional Neural networks’, in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, Presented at the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), 1091–1095.

Appendix B: Code Listing

```
#Importing Libraries
from PIL import Image
import numpy as np
import sys
import os
import csv
from keras.models import Sequential
from sklearn import preprocessing
from keras.utils import np_utils
from keras.layers import Dense, Dropout, GaussianNoise, Conv1D
from keras.preprocessing.image import ImageDataGenerator
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
```

Code Listing 1 Importing the necessary libraries

```
# Converting all the images from JPG to CSV
filelist = []
for path in [x[0] for x in os.walk('D:/thesis_dataset/')][1:]:
    label = os.path.basename(path)
    for root, dirs, files in os.walk(path, topdown=False):
        for name in files:

            if name.endswith('.jpg'):
                fullName = os.path.join(root, name)
                filelist.append([fullName,label])
row = []
for file,label in filelist:

    print(file)
    img_file = Image.open(file)


    # get original image parameters...
    width, height = img_file.size
    format = img_file.format
    mode = img_file.mode

    # Make image Greyscale
    img_grey = img_file.convert('L')

    # Save Greyscale values
    value = np.asarray(img_grey.getdata(), dtype=np.int64).reshape((img_grey.size[1], img_grey.size[0]))
    value = value.flatten()
    row.append([value,label])
```

Code Listing 2 Converting the image to .csv format

```
# Display the images in the dataframe
labels = []
arrays2 = []
for arr in row:
    labels.append(arr[1])
    arrays2.append(arr[0])
df = pd.DataFrame(arrays2)
df['label'] = labels
df.head()
```

Code Listing 3 Preparing the data frame

```
#Encode the label column
pixels = df.drop(["label"],axis=1)
label = df["label"]
le = preprocessing.LabelEncoder()
encoded_label = le.fit_transform(label)
```

Code Listing 4 Encoding the label column

```
#Function to display the images using the points in each column
def show_images(pixels):
    fig, axes = plt.subplots(5, 6, figsize=(11, 7), subplot_kw={'xticks':[], 'yticks':[]})
    for i, ax in enumerate(axes.flat):
        ax.imshow(np.array(pixels)[i].reshape(250, 250), cmap='gray')
    plt.show()
```

Code Listing 5 Defining a function to display the images

```
x_train, x_test, y_train, y_test = train_test_split(pixels, encoded_label)
```

Code Listing 6 Splitting the dataset

```
# Extracting feature using PCA
NCOMPONENTS = 15

pca = PCA(n_components=NCOMPONENTS)
X_pca_train = pca.fit_transform(X_sc_train)
X_pca_test = pca.transform(X_sc_test)
pca_std = np.std(X_pca_train)

print(X_sc_train.shape)
print(X_pca_train.shape)
print(y_train.shape)
Y_train = y_train.astype('int32')
Y_train = np_utils.to_categorical(Y_train)
print(Y_train)
```

Code Listing 7 Feature Extraction using PCA

```
#pca = PCA(n_components=NCOMPONENTS, svd_solver='full').fit(x_train)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance');
plt.show()
```

Code Listing 8 Cumulative variance graph representation for PCA

```
def show_eigenfaces(pca):
    fig, axes = plt.subplots(3, 5, figsize=(9, 4),subplot_kw={'xticks':[], 'yticks':[]})
    for i, ax in enumerate(axes.flat):
        ax.imshow(pca.components_[i].reshape(250, 250), cmap='gray')
        ax.set_title("PC " + str(i+1))
    plt.show()
```

Code Listing 9 Defining a function to show the feature extraction for PCA

```
#Extracting feature using LDA
lda = LDA(n_components=3)

X_lda_train = lda.fit_transform(X_pca_train,y_train)
X_lda_test = lda.transform(X_pca_test)
lda_std = np.std(X_lda_train)
print(X_sc_train.shape)
print(X_lda_train.shape)
print(y_train.shape)
Y_train = y_train.astype('int32')
Y_train = np_utils.to_categorical(Y_train)
print(Y_train)
```

Code Listing 10 Feature Extraction for Fisherface

```
plt.plot(np.cumsum(lda.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance');
plt.show()
```

Code Listing 11 Cumulative variance graph representation for Fisherface

```
model = Sequential()
layers = 1
units = 128
pca_std = np.std(X_pca_train)
#Dense layer with 128 neurons
model.add(Dense(units, input_dim=NCOMPONENTS, activation='relu'))
#regularisation layer
model.add(GaussianNoise(pca_std))

#We are adding only 1 set of layer with 128Neurons
for i in range(layers):
    #Dense layer
    model.add(Dense(units, activation='relu'))
    #Both below for regularisation
    model.add(GaussianNoise(pca_std))
    model.add(Dropout(0.1))

#Output layer ; 5 because there are only 5 labels, "softmax" is used because it has multiple labels.
model.add(Dense(5, activation='softmax'))
model.summary()


model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['categorical_accuracy'])

history = model.fit(X_pca_train, Y_train, epochs=100, batch_size=256, validation_split=0.15, verbose=2)
```

Code Listing 12 Model Training for PCA

```
predictions = model.predict_classes(X_pca_test, verbose=0)
predictions1 = le.inverse_transform(predictions)
print(classification_report(y_test,predictions))

def write_predictions(predictions1, fname):
    pd.DataFrame({"ImageId": list(range(1,len(predictions1)+1)), "Label": predictions1}).to_csv(fname, index=False, header=True)

write_predictions(predictions1, "pca-keras-mlp.csv")
```

Code Listing 13 Prediction for PCA

```
model = Sequential()
layers = 1
units = 128
lda_std = np.std(X_lda_train)
model.add(Dense(units, input_dim=3, activation='relu'))
model.add(GaussianNoise(lda_std))
for i in range(layers):
    model.add(Dense(units, activation='relu'))
    model.add(GaussianNoise(lda_std))
    model.add(Dropout(0.1))
model.add(Dense(5, activation='softmax'))
model.summary()

model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['categorical_accuracy'])
history = model.fit(X_lda_train, Y_train, epochs=100, batch_size=256, validation_split=0.15, verbose=2)
```

Code Listing 14 Model Training for Fisherface

```
predictions = model.predict_classes(X_lda_test, verbose=0)
predictions1 = le.inverse_transform(predictions)
print(classification_report(y_test,predictions))

def write_predictions(predictions1, fname):
    pd.DataFrame({"ImageId": list(range(1,len(predictions1)+1)), "Label": predictions1}).to_csv(fname, index=False, header=True)

write_predictions(predictions1, "lda-keras-mlp.csv")
```

Code Listing 15 Prediction for Fisherface