

Practical Face Recognition with the Raspberry Pi

by

Muhammad Talha Bin Ijaz

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

August 2021

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.



Muhammad Talha Bin Ijaz

August 29, 2021

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.



Muhammad Talha Bin Ijaz

August 29, 2021

Acknowledgments

I would like to thank my dissertation supervisor, Professor John Waldron for his invaluable support and guidance on the undertaken project. Moreover, I would like to extend this gratitude to all of the professors and faculty of the university for their guidance and help over the past year. Lastly, I want to thank the class of 2021 for the MSc in computer science at Trinity college dublin for the memorable year.

Muhammad Talha Bin Ijaz

University of Dublin, Trinity College
August 29, 2021

Abstract

Face recognition is one of the most studied research problems in the pattern recognition problem family. It refers to the problem of detecting and recognizing human faces in digital images. It has a wide range of applications from identity management, access control, and behaviour analytics along with more controversial applications such as automated public surveillance. This dissertation studies the field of face recognition in context of a specific real world application, namely, identity verification in access control systems for office buildings and other places of work. The dissertation starts off with a general overview of the problem and discusses the individual components of face recognition such as face detection, facial embeddings and face recognition. The pertaining theoretical ideas and the current research landscape is studied for each of these components. The second section of the dissertation is experimental and studies the effectiveness of popular algorithms used in face recognition such as viola jones, hog detectors and deep learning. Experiments are carried out on several face detection and recognition datasets to measure metrics such as speed, accuracy and overall performance on a raspberry pi. The goal is to pick the best algorithms in terms of accuracy and speed for the individual components that can be used to construct an end to end, near real time face recognition pipeline. The last part of the dissertation deals with the development of a plug and play face recognition platform that can be used by office buildings to add face recognition to their access control system . This platform consists of a web application that can be used to manage multiple face recognition nodes. The features of this platform include device registration, employee registration, attendance reports, employee access management and alerts for unauthorized access attempts. In addition to the web platform, a local application is developed for a raspberry pi in order to convert it into a face recognition node that works in conjunction with the web application to provide a seamless experience. The goal of the developed consumer system is to provide an easy to set up, batteries included solution to incorporate face recognition as an added layer of security.

General Information

All of the implementation code and the analysis notebooks can be found in the attached zip file or at the following github link:

Github Repo: <https://github.com/talhabinijaz576/dissertation>

The overall repository is divided into three folders. The “*experiments*” root folder pertains to the raw python implementations and the analysis notebook. Moreover, the “*rpi_app*” folder contains the desktop application developed for the raspberry pi. Lastly, the “*server*” folder contains the django web application that contains the webportal and the websocket server. For your convenience, the web application has been made live and is accessible at the following location.

Url: <https://facerecognition.jazeetech.com/>

Username: test

Password: test123

Word Count: 27,263 words

Table of Contents

Section 1 - Introduction	9
1.1 Motivation	9
1.2 Objectives	10
Section 2 - The Face Recognition Pipeline	12
2.1 Face Detection	12
2.2 Face Alignment	13
2.3 Facial Embedding	13
2.4 Face Verification	14
2.5 Face Identification	14
Section 3 - Face Detection	15
3.1 Performance Evaluation	15
3.2 Datasets	18
3.2.1 WIDER Face dataset	18
3.2.2 Face Detection Dataset and Benchmark (FDDB)	19
3.3 Algorithms	20
3.3.1 Viola and Jones (Haar Cascades)	21
3.3.2 Histogram of oriented gradients (HOG)	23
3.3.3 Tiny YOLO V3	25
3.3.3 MobileNet Single Shot Detector (SSD)	26
3.4 Experimental Results	29
3.4.1 Methodology	29
3.4.2 Results	30
3.4.3 Number of Faces	33
3.4.4 Face to Image Ratio	34
3.4.5 Conclusion	36
Section 4 - Face Alignment	38
4.1 Performance Evaluation	39
4.2 Datasets	41
4.3 Algorithms	42
4.3.1 Ensemble Randomized Trees (ERT)	43
4.3.2 Multi task Cascaded Convolutional Neural Network	44
4.4 Experimental Results	46
4.4.1 Methodology	46

4.4.2 Results	48
Section 4 - Face Recognition	51
4.1 Performance Evaluation	52
4.2 Datasets	53
4.2.1 Labeled Faces in the Wilds (LFW)	53
4.2.2 Cross Age Celebrity Dataset (CACD)	54
4.3 Algorithms	55
4.3.1 Facenet / OpenFace	57
4.3.2 MobileFacenet	58
4.3.3 ArcFace	59
4.3.4 CosFace	60
4.3.5 VGG-Face2	62
Section 5 - Face Verification	63
5.1 Evaluation Methodology	63
5.2 Threshold Selection	64
5.3 Initial Results	65
5.4 Impact of Face Size	67
5.5 Impact of Age and Difference in Ages	68
5.6 Performance Evaluation for Different Genders	71
5.7 Performance Evaluation for Different Ethnicities	72
5.8 99.9% True Positive Rate	74
5.9 Conclusion	76
Section 6 - Face Identification	77
6.1 Evaluation Methodology	78
6.2 Threshold Selection	78
6.3 Initial Results	79
6.4 Impact of Dataset Size	82
6.5 Dynamic Thresholding	85
6.5 Conclusion	86
Section 7 - Web Portal and Desktop App	87
7.1 Web Portal	87
7.2 Desktop Application	93
7.3 One Click Install	93
Section 7 - Conclusion	94
Bibliography	96

Section 1 - Introduction

Face recognition is a biometric verification technology that involves locating, extracting and identifying a human face by matching it against a dataset of faces. Face recognition works by identifying face landmarks and features and uses a comparison mechanism to compare and match human faces from a still image or a video frame. The earliest form of face recognition was developed in the early 1960s and worked by pinpointing the coordinates of manually selected, crude features from images of faces. Since then, the field has come a long way and is considered one of the most popular and well researched sub fields in artificial intelligence and computer vision. Face recognition technologies have a wide array of commercial and security applications such as identity management, access control, behaviour analytics and public surveillance.

Several major achievements have been made in the field over the past few decades that have propelled the performance of face recognition to human levels. Present day face recognition algorithms are extremely robust and have achieved satisfactory performance in complex, unconstrained environments. Moreover, the exponential increase in processing capacity has made face recognition technologies viable for practical, real time applications. The non-intrusive nature and the lack of custom hardware requirements such as iris scanners has made face recognition a front runner in modern biometric systems. Today, face recognition is adopted at a large scale due to its contactless nature despite a lower accuracy compared to fingerprints verification or iris verification.

1.1 Motivation

Face recognition is arguably one of the most active areas of research in the computer vision field. There is an incredibly large amount of research in the field with new algorithms and techniques launching by every passing week. The overwhelming amount of research over the past decades makes it difficult for new people to break into the fields. In addition to this, face recognition is more complex of a problem compared to vanilla object recognition or detection problems as it consists of multiple distinct components which adds up the difficulty. One of the primary motivations for this dissertation is to offer a condensed and linear view of the face recognition landscape with the goal of providing a comprehensive overview of each phase of the face recognition pipeline.

Even with a deeper understanding, the amount of algorithms in a single phase is overwhelming and comparative analyses are hard to come by in the research literature despite a multitude of papers that discuss individual algorithms separately. The second motivation for this dissertation is to select the best candidate algorithms for each phase and compare them within a common context. An in depth analysis is required to not only compare the overall performances of the selected algorithms, but also to study the behaviour of the algorithms in different circumstances. In favour of condensation, the goal is to conclude with the selection of a single algorithm for each of the phases for people who do not require an in-depth understanding.

Still, it is hard to use it in commercial applications due to the complicated supporting infrastructure and applications that enable non-technical people to utilize the technology. The third and final motivation for this dissertation is to implement an end-to-end plug and play face recognition system with a “batteries included” approach. This includes not only an easy to use implementation of the face recognition pipeline, but also the supporting infrastructure such as a centralized webportal, desktop application, raspberry pi OS images and on click installation scripts. This is focused on non technical people who are not programmers and have no need to understand the inner workings of a face recognition system. A real world application is developed that can be configured in minutes without any real programming knowledge and serve as a complete solution for adding face recognition as an additional security measure to office buildings.

1.2 Objectives

The formal objectives of this dissertation are closely related to motivations mentioned in the previous sub-section. The dissertation hopes to accomplish four distinct objectives:

- Identify and explain the individual phases that makeup the face recognition pipeline;
- Discuss historical and modern algorithms for each component and select several candidates for each phase for further testing. Finally, a detailed

literature review is to be done that explains the inner workings of the algorithms;

- Implement the selected algorithms in a programming language. Then, design and execute comprehensive experiments to perform a comparative analysis for the algorithms in different situation and from different perspectives with the goal of choosing a single best algorithm for each phase of the pipeline that can be used in a real world application;
- Develop a complete, plug and play face recognition solution that can be easily set up and used by the layperson for employee access control in office buildings. The solution should abstract away the technicalities of the underlying system and should be accessible to the average person. The goals for this system are as follows:
 - Centralized control room (web application) for management;
 - Real time server for real time communications with desktop app;
 - Autonomous desktop application to be run on a raspberry pi;
 - Preconfigured raspbian OS images for the raspberry pi;
 - One click registration scripts to connect nodes to the control room.

Section 2 - The Face Recognition Pipeline

The section serves as an overview of the thesis with a goal of explaining the background of the different concepts in face recognition. First, the main components of a face recognition pipeline are discussed individually as well as in the context of a complete, end-to-end pipeline. Moreover the experiments carried out to evaluate the performance of the various algorithms used for each of the components of the pipelines are introduced and discussed.

The broader face recognition pipeline can be broken down into several components. These components include face detection, face alignment, calculation of face embeddings, face verification and face recognition.

2.1 Face Detection

Face detection constitutes the first phase in the end-to-end face recognition pipeline. Before performing any comparison or “recognition”, it is necessary to process the image provided in order to identify and localize all of the faces (if any). Face detection refers to the problem of identifying and location human faces in a given picture. The goal of face detection is two-folds. The first goal is to determine how many, if any, faces are present in a given picture and the second goal is to locate and determine a bounding box for each individual face. There has been an extensive amount of research in face detection and today, there are a lot of traditional algorithms as well as deep learning algorithms that are used for face detection. These four most popular algorithms, the Viola Jone algorithm, the Histogram of Oriented Gradients (HOG) detector, the Yolo V3 Tiny convolutional neural network and the MobileNet single shot detector (SSD) convolutional neural network are discussed in detail in section 3. An explanation is provided for each of them to illustrate how they work and a complete experimental analysis is performed on a raspberry to compare the algorithms, evaluate their performance and determine the best face detection algorithm to be used in our application.

2.2 Face Alignment

After detection and localization, the face must be conformed to a common standard in order to effectively compare it with other faces for verification or recognition. Face alignment is necessary due to the existence of different human poses and face expressions as a robust face recognition application should be, for example, able to compare a fully front facing human face with the face looking sideways. Face alignment is typically done by identifying landmarks such as eyes, nose, lips etc on the detected face and using the facial landmarks as reference points to transform the image perspective. Traditional alignment techniques involve using a landmark detector to locate the face and using computer vision algorithms such as perspective transform for the alignment. Recently, however, a lot of deep learning algorithms have emerged that replace the two phase alignment process with end to end face alignment.

2.3 Facial Embedding

A technique to convert faces into some sort of mathematical representation is necessary in order to compare two faces and evaluate their similarity. Face embeddings are high quality features that are extracted from images of aligned faces in the form of mathematical vectors. Older techniques for extracting the face embeddings include traditional feature extractors such as hog features while the newer techniques include using the feature extractor component of general convolutional neural networks or custom deep learning models such as FaceNet that are trained specifically for face recognition. The face embeddings naturally relate to the face verification and recognition components of the pipeline and are considered the most crucial part of a face recognition system. The most widely used face embedding techniques are explained, evaluated and compared in section 4 of the dissertation.

2.4 Face Verification

Face verification refers to the part of the face recognition pipeline that deals with the comparison of two faces in order to determine the similarity. It is a binary classification problem in which the output is either “yes” if the faces belong to the same human or “no” if they belong to different people. Face verification is carried out by comparing the “distance” between the respective numerical embeddings (mathematical representation vectors) of the two faces and assigning the pair a verification score. A threshold that is determined through trial and error is used to make the final judgement. The verification is considered successful if the verification score (the distance) is higher than the threshold and a failure otherwise.

2.5 Face Identification

Face identification is the generalized form of face verification. It refers to the problem of recognizing the face of a human from a predefined collection of human faces. The outputs of face recognition are two folds. The first goal is to determine if the face in question is recognized (that is, matches one of the faces in the collection) or is unknown. The second goal is the determination of the person, if any, the face belongs to. This is typically the last component of the pipeline and does not require complex algorithms of its own. It is typically carried out by verifying the new face with every face in the database and checking the verification score. If the verification score of the highest performing face is above a certain threshold, the recognition is deemed successful and the identity of the person is returned as output. The recognition is deemed a failure and the face is marked as unrecognized if the highest score is below the threshold.

Section 3 - Face Detection

Face detection refers to the problem of identifying and location human faces in a given picture. Face detection constitutes the first phase in the end-to-end face recognition pipeline. The goal of face detection is two-folds. The first goal determines how many, if any, faces are present in a given picture. The second goal is to locate and determine a bounding box for the face.

A bounding box can be elliptical (with a center coordinate, width and height of the oval) or rectangular box with four coordinates that correspond to the four corners of a rectangle. There are merits and demerits for either choice and the issue has been researched extensively with rather inconclusive results. While the theoretical characteristics may be comparable, rectangular bounding boxes are significantly easier to define and compare to each other. Furthermore, rectangular boxes are more explicit which makes them a better choice for machine learning algorithms. Finally and most importantly, most of the public face detection and recognition datasets consist of rectangular bounding boxes so in order to perform a more comprehensive analysis, this dissertation will only deal with evaluation strategies, datasets and analysis with rectangular bounding boxes.

To summarise, a face detection workflow takes a single picture as an input and returns a list of bounding boxes where each bounding box comprises a set of points that locate a single face.

3.1 Performance Evaluation

A perfect face detector would be able to identify every face present in an image without giving any false positives (results where no human faces are found) and locate them perfectly. Hence, the evaluation of a face detection algorithm can be broken down into two components.

The first component of the evaluation corresponds to the collective performance of the algorithm on all faces. It is a discrete classification problem and refers to the proportion of faces correctly identified. Before we go any further, it is important to establish the exact definition of what constitutes a correct identification. Since the result is continuous as it constitutes the coordinate of a bounding box, a threshold must be set to discretize the results, associate each bounding box with an individual face and evaluate false positives and false negatives. The general consensus among the research community is to use some form of overlap ratio between the prediction and the ground truth to execute the evaluations. The most common metric used is the “Intersection over Union (IoU)” which is the ratio of overlap of a predicted bounding box B1 and the ground truth bounding box (B2). It can be further illustrated as follows:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


For evaluation, the IoU value for every possible pair of the prediction boxes and the ground truth boxes. The IoU value ranges from 0 to 1 with 0 representing “no overlap” and 1 representing “perfect overlap”. For our purposes, we can set the threshold at a reasonable value such as 0.5. This means, if a predicted bounding box is associated with a ground truth bounding box if the IoU value is greater than 0.5. All ground truth bounding boxes without an associated predicted bounding box are labelled as false negatives while all predicted bounding boxes that remain unassigned are labeled as false positives. The aforementioned perfect face detector would have a strictly one-to-one association between the predicted bounding boxes and the ground truth with zero unassociated predicted boxes (no false positives) and zero unassigned ground truth bounding boxes (no false negatives).

Once this is done, a single metric can be calculated for the collective performance (or overall accuracy) for a particular image. A naive metric would be simple accuracy or the percentage of ground truth boxes correctly predicted. It, while simple, ignores the false positives and would incentivize the final system to simply predict as many faces as possible. A better approach would be to consider the

precision and recall metrics that take in account the overall performance, the number of false positives and the number of false negatives.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad \text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

In addition to the precision and recall metrics, it is worth evaluating how “correct” the predicted bounding boxes are with reference to the associated ground truth bounding boxes. This score is readily available at this point in the form of the IoU score. In order to obtain a singular metric for a particular image, a simple mean of the IoU scores of all the correctly associated pairs can be taken.

$$p = \frac{1}{n} \sum_{i=1}^n \text{IoU}$$

Finally, we have to consider the time requirements and performance constraints that a raspberry pi poses. Since the entire face recognition pipeline has to be run in real time, it is vital that the face detection component does not take more than a few milliseconds in order to achieve a decent frame rate. All experimental tests should be done on a raspberry pi for this reason and the average time per image should be measured for each of the algorithms.

In order to compare two face detection algorithms, the Receiver Operating Characteristic (ROC) curve which takes in account both precision and recall along with mean IoU metric can be compared separately in context of the time requirements and hardware constraints of a raspberry pi. Since the number of face detection algorithms is limited, the best choice can be selected manually. Using the evaluation approach and comparison strategy mentioned in this section, it is possible to select the best face detection algorithm that correctly identifies the highest number of human faces with a minimum number of false negatives, highest precision that works within a reasonable amount of time.

3.2 Datasets

Tests and experiments have to be done in order to evaluate and compare the performance of the face detection algorithms. These tests are carried out on several publically available datasets. It is important to select the appropriate datasets that reflect the settings of the final system as closely as possible. As mentioned above, the consumer system designed and implemented in this dissertation is supposed to be used for access control and employee verification inside office buildings. The indoor settings somewhat simplifies the problem as the variation in the images that the system can be seen is relatively constrained. That said, the selected datasets for the tests are to be chosen to resemble in door settings as much as possible.

The following subsections deal with the several face detection datasets that are used to run the experiments:

3.2.1 WIDER Face dataset

One of the most interesting face recognition datasets I found is the WIDER face dataset. This dataset was made public in correspondence with the research paper in 2016 and is by far, the most diverse, easily accessible dataset available in public. The dataset comprises over 32,000 individual images that contain a large number of faces each. Face annotations are done manually and the recognizable faces where at least 30% of the face is visible. All of the faces in images are annotated with rectangular bounding boxes which is ideal.

A casual inspection of the dataset reveals that pictures are taken from further away which closely resembles the office environment as the mean distance from the camera to the room is similar to the distance between the camera and the faces in the dataset. Not only that, each image contains, on average, 12 faces which is similar to what we would expect in a single room in an office setting. Furthermore, the general theme seems to be unconstrained scenes as there is a huge variation in the images which may help us estimate performance in unconventional office settings such as warehouses where access control is more relevant. In addition to this, most of the images of the people are full body and do not just focus on upper

bodies or closeups. These features make this data set an ideal testing ground to evaluate and compare the performance of different face detection algorithms.

The dataset also segregates the images in 61 event categories that correspond to the background setting of the image. This means that the appropriate categories that are similar to a workplace setting can be selected to further reduce the number of images and increase the relevancy to the problem at hand.. Based on a cursory review, I chose, handshaking, press conference, meeting, group, interview and greetings category to name a few. Out of the original 32,000 images, the final refined dataset comprises around 2700 images from sixteen categories which is large and diverse enough to achieve a statistically sound conclusion and small enough to run the experiments in a reasonable amount of time.

3.2.2 Face Detection Dataset and Benchmark (FDDB)

Another promising dataset is the face detection dataset and benchmark (FDDB). While somewhat dated at 12 years old, the images still look very decent. The dataset contains about 3000 images collected from the Yahoo news website and seems to be unconstrained in nature as it contains images of different dimensions, perspectives and event settings. Since the average number of faces per image is low, obstructed and partial faces are not a big problem and the annotations are very clear. Both elliptical and rectangular bounding boxes are available for each face.

Compared to the WIDER face dataset, the images seem to be taken from a lot closer up. Furthermore, most images are very candid in nature and tend to focus on the upper torso rather than the full body. In addition the images contain between 1 and 3 faces per image so it may not be 100% relevant to our place settings. On the other hand, the images do contain several scenarios, backgrounds and situations which provides enough diversity. No event categories are found so it is not possible to further reduce the size of the dataset.

My primary rationale for selecting this dataset was to provide a different perspective. The differences in the FDDB and the WIDER dataset can be used to

highlight the contrasts in the results and to compare the face detection algorithm in different situations (for example close up vs far away or single face vs large number of faces).

3.3 Algorithms

Face recognition is one of the most widely studied topics in artificial intelligence with the earliest research dating back to the early 1970s when Woody Bledsoe and Helen Wolf developed a rudimentary mathematical methodology to compare faces. Since then, we have come a long way to sophisticated convolutional neural networks with billions of parameters that outperform humans on pretty much all fronts. The recent development in machine learning and statistical methods have allowed for more robust face detection system that can manage

Even though the earlier research is very relevant from a historical and mathematical point of view, this dissertation mainly focuses on more recent algorithms including but not limited to deep learning convolutional neural networks. The most important breakthrough in modern face detection is definitely the Viola and Jones algorithm released by Paul Viola and Michael Jones in 2001. The viola and jones algorithm was the first algorithm with performance comparable to that of humans as it achieved a 75% to 90% detection rate on various face detection datasets. In addition to the Viola and Jones algorithm, the Histogram of Oriented Gradients (HOG) algorithm is also widely popular. In recent years the majority of the face detection research occurs in the realm of deep learning due to the exponential increase in processing power. Deep learning is a family of blackbox algorithms inspired by the biological neural networks. Deep learning is used for complex prediction tasks where manual feature crafting is too complicated. Members of this family include the multilayer perceptrons (MLP), artificial neural networks, recurrent neural networks, LSTM and convolutional neural networks.

Convolutional neural networks are, by far, the most popular choice for object detection tasks so only they are discussed and experimented with in this

dissertation. CNNs are end to end algorithms that require little to no preprocessing and take care of everything from feature generation, feature selection and the actual classification. They exploit the structural information of the image by recursively applying filters (or kernels) in a layered manner. The initial layers extract simpler features such as geometric shapes while subsequent layers build on those features to generate more complex features. The last convolutional layer of the network can extract very complex features such as an entire eye from a face. Final layers of the CNNs are typically fully connected and use the extracted features to make a final classification. Convolutional neural networks can be further divided into object recognition networks and object detection networks. The aim for object recognition is to simply recognize the image and predict the object class. On the other hand, object detection deals with object recognition as well as localisation which is why only the second category is appropriate for face detection.

Unfortunately, the raspberry pi is not able to run the majority of deep learning convolutional neural networks due to resource constraints so we can only consider the least complex, mainstream models such as the YoloV3-tiny and the Single Shot MultiBox detector (SSD).

3.3.1 Viola and Jones (Haar Cascades)

The Viola and Jones algorithm was first released by Paul Viola and Michael Jones in 2001. The significance of this algorithm comes from the high accuracy (for the time) as well as being the first real time face detection algorithm. Briefly speaking, the viola jones algorithm uses running a sliding window over the image and extracts some image features. These features are then run through a classifier to classify faces from non-faces. The algorithm comprises four steps. There are two steps pertaining to model training and two for the actual face detection. In the training phases the classifiers must be trained and the best features must be selected whereas in the detection phase, haar-like features are detected and an integral image is calculated.

The first main concept is the Haar-like features similar to Haar wavelets. Haar-like features exploit the localized differences and contrast to extract image features by exploiting the difference in the sum of grayscale pixel values. For example, a vertical edge can be detected using a haar function that exploits the difference in the sum of pixel values on the left side and the right side of the box. The three main types of Haar-like features proposed by the authors are edge features designed to detect edges, line features designed for line segment detection and four side features. A high value for a haar feature means a higher probability of the existence of the image characteristic it represents. Thresholding is applied to filter the lower values and determine if the feature exists.



Example of Haar Features

Calculations with a sliding window are very computationally expensive, however, Viola Jones does this very efficiently using integral images, also called a summed area table that do the calculation in one go. A value for each coordinate (x, y) is calculated by summing all values to the left and above. Then, the individual sums within a rectangle are calculated by referencing the summed area table and performing simple arithmetics using the point values for the four corners, thus emulating the sliding window effect mentioned above.

The algorithm uses multiple iterations of adaptive boosting (AdaBoost) to select and filter the overwhelming number of features in order to create an effective classifier. The model tests each feature individually and ranks them on the number of false positives and false negatives in the predictions. For example a feature that exists in 4 face images and 1 non face image is considered to be better than a feature that exists in 3 face images and 2 non-face images. Once the best feature is selected, the next features are selected based on how well they complement the previous best feature by assigning greater importance to the false negatives and using a weighted cumulative function ensuring a collective outlook. With this

approach, it is possible to select just 200 of the most important features that best complement each other.

The final piece of the puzzle is known as “Cascading” Instead of using a single classifier, a cascade of classifiers is used to speed up the model and increase the accuracy. Early in the cascade, simple models (such as the existence of the most important feature) are used that immediately reject the vast majority of subwindows within the image. Classifiers then become increasingly complex at every step for more careful selection to reduce false positives. This “cascade” of simple and complex classifiers ensures that no time is wasted on the majority of the image subwindows where no faces are found.

Viola Jones is a promising algorithm that is still used for face detection where computing resources are constraints which makes it the ideal candidate for a raspberry pi. It is efficient and works well in a large environment. On the other hand, Viola Jones was developed for the frontal view of faces so the algorithm works best with pictures where people face the camera instead of looking sideways. Partial faces and unusual positions do not bode well with the algorithm. This may not be a problem as face recognition systems in buildings have people looking directly. Still, the algorithm is tested due to its reputation and efficiency. A complete evaluation and the results of tests can be found in the experimental section of the thesis.

3.3.2 Histogram of oriented gradients (HOG)

Histogram of oriented gradients (HOG) detector was proposed by Naveet Dala and Bill Triggs in 2005 as a better, more robust alternative to the Viola Jones algorithm. Like Viola Jones, HOG detectors work by first extracting images features and then running a classifier to determine the existence of a face. Unlike Viola Jones, this algorithm uses histogram features generated using the edge gradients (small changes in spatial directions) of the image instead of haar features and a support vector machine (SVM) classifier which leads to more robust face detection. The

detector works well, is fast enough to be run on a raspberry pi and remained the state of the art for face detection until the deep learning era.

The feature extraction is based on the evaluation of localized histograms of gradient orientation. The premise is that the shape and appearance of the object is well reflected in the gradient intensity of the “direction” of the edges. The original image is broken in a grid of regions. The original image is broken down in a localized region and the gradients and orientations are calculated using the “slope” of the pixel values of adjacent pixels. The value of the gradient is higher whenever there is a greater change in pixel intensity as is the case around edges. These gradients and orientations are then used to generate a histogram that is essentially a frequency distribution of a set of adjacent points in the region. This can be done by binning the gradients and orientations and calculating the number of instances in each bin. A histogram is calculated for a 8x8, 16x16 or 32x32 image region. A sliding window is imitated where the stride is shorter than the width which means there is significant overlap between regions. The region histograms are then combined to generate feature vectors that are further normalized to account for spatial differences. Finally, the feature vectors can be used to make final predictions. The authors recommend a 64x128 detection window for the feature extraction and a soft linear support vector machine (SVM) classifier with a for the predictions. A gaussian kernel has a slightly better performance but is forgone for performance reasons.

Hog detectors have several benefits. They capture the edges and the structure of the image very accurately which works well for distinct shapes such as faces. Furthermore, the features are scale invariant and can accommodate a wide variety of distortions and resolutions. This also means the HOG detectors can manage more perspectives compared to a Viola Jones detector. That said, HOG detectors are still meant for frontal faces and faces looking sideways hamper the face detection performance. According to initial research on the HOG detectors, it seems that they might be on the sweet spot between the fickle Viola Jones and computationally expensive convolutional neural networks. The final decision is made after the evaluation but it seems to be the most promising candidate for the face detection component of our pipeline.

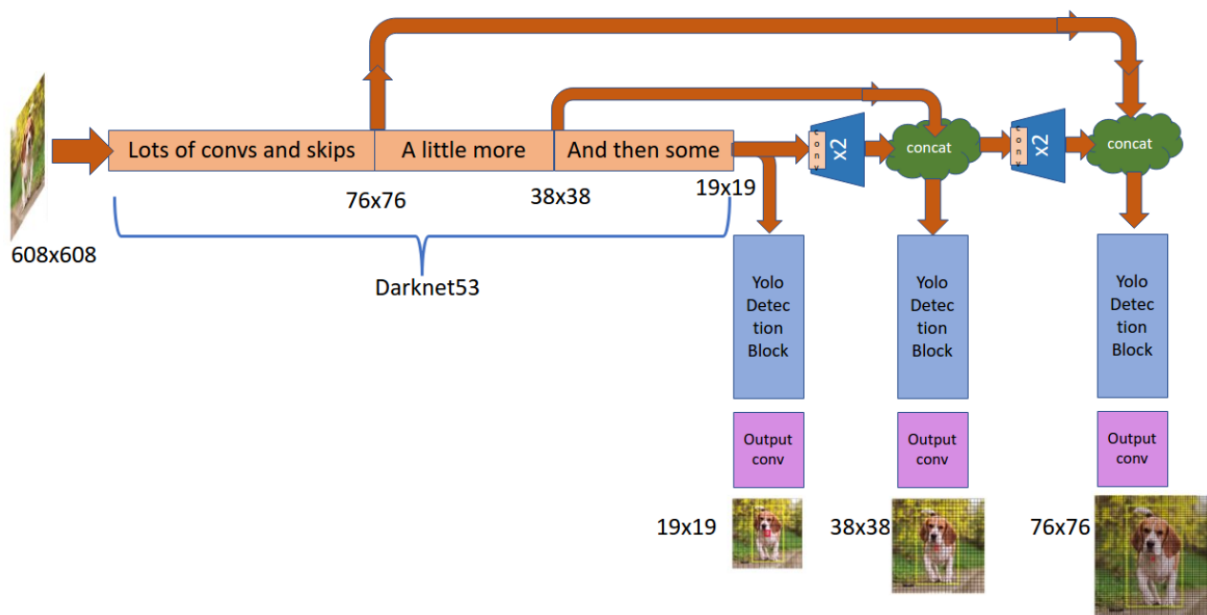
3.3.3 Tiny YOLO V3

Most of the recent research in face detection focuses on deep learning. Convolutional neural networks are considered the state of the art in object detection. Unfortunately, the planned system is built around a raspberry pi which is too weak to run a convolutional neural network. An example would be the state of the art R-CNN networks. While they outperform humans, a single pass takes upwards of 10 seconds on a raspberry pi which is unacceptable. Therefore, I picked one of the fastest object detection models called the YoloV3 which needs a fraction of the resources required by traditional networks.

Yolo stands for “you only look once” as the model is based on a new approach that unifies the feature extraction, object classification and object localization into a single component. YoloV3 is the third iteration of the popular yolo object detection algorithm family and comes in two flavours, the standard model and the more compact tiny model. Research in the Yolo V3 family indicates that only the tiny version would be able to run in reasonable time on a raspberry pi. In addition to lesser complexity, the network architecture and optimization functions of the tiny version of YoloV3 is also optimized to be run on mobile devices that makes it more suitable for our application. As hinted in the name, the neural network is based on a single stage architecture with the goal of executing the prediction in a single pass and reducing the redundant calculation to the absolute minimum in order to significantly speed up the prediction process.

The network is made of 13 convolutional layers with 16 to 1024 filters each. In addition, the network uses 7 maximum pooling layers along with the relu activation and upsampling for better performance. The network does not contain fully connected layers which means it is very flexible with different resolutions and aspect ratios. The network architecture also includes skip connections that allow the features extracted to be directly carried forward. This means that instead of a sequence, some intermediate features are allowed to skip some of the subsequent layers. For example, extracted by a kernel in layer 5 can be carried forward to layer 9 without having to go through all of the layers in the middle. This is done to ensure flexibility and allows the network to directly use earlier, simpler features

without adding complexity. The network also has multiple prediction heads that each make a separate prediction at different intervals of the network and process the image at different compression levels. This is done to increase the robustness of the network by making it scale invariant and to allow it to learn about objects at different sizes.



Compared to other mainstream convolutional neural networks, the Tiny YOLO V3 has a much smaller footprint but is comparable in performance. The main drawback is the complicated training process and limited number of object categories. The training complexity is not a problem because the trained network weights are freely available online eliminating the need for retraining and the limited object classes are not a concern as we are only interested in a single object category making the Tiny YoloV3 the best candidate for our implementation.

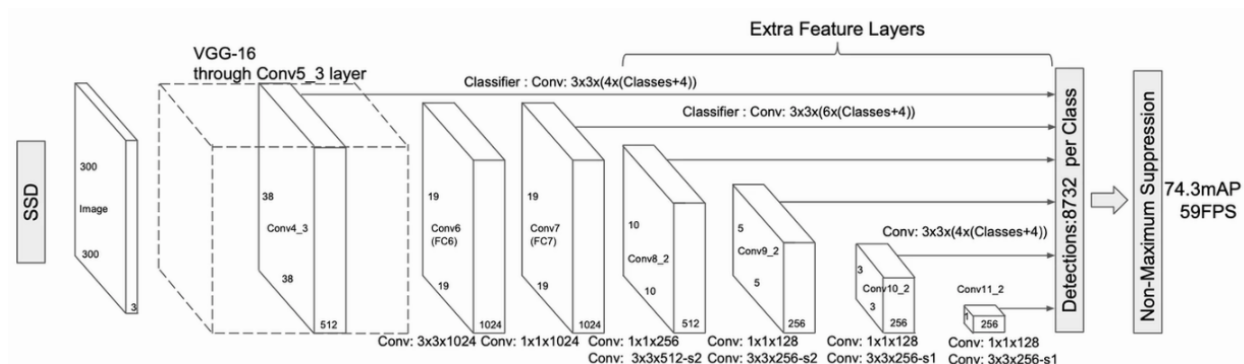
3.3.3 MobileNet Single Shot Detector (SSD)

Another promising deep learning candidate for face detection is the MobileNet single shot detector. Released in 2016, the MobileNet SSD was one of the first

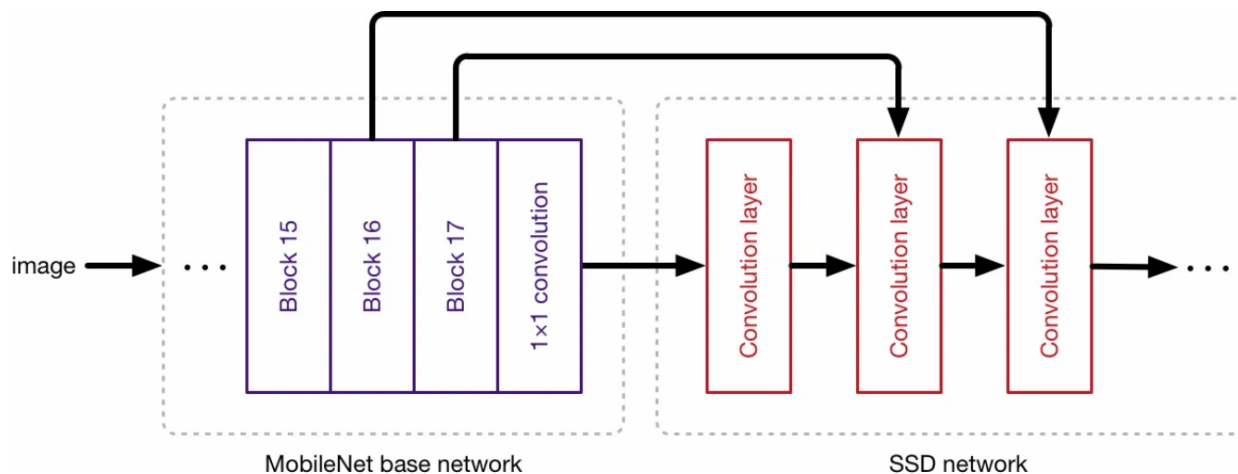
significant ventures in real time object detection on embedded devices (such as a raspberry pi) without sacrificing accuracy. This network is made up of a combination of mobile net, which is a lightweight object recognition network and SSD that is an object detection network. Like yolo, the goal of single shot detectors is to process the image in a single take in order to minimize redundant calculations and to achieve the speed boost.

The first part, MobileNet, is made up of 23 convolutional layers with 32 to 512 convolutional filters each. The network also uses average pooling layers between the convolutional layers to condense intermediate features along with the ReLU activation function. Moreover, batch normalization is used after each convolutional layer to normalize the layer output for quicker, more stable learning. The network has one final fully connected layer and one hot softmax activation for the final classification. Unlike yolo, the mobile net has a fixed input shape of 224x224 due to the fully connected layer. What sets the mobile net apart is the addition of depth-wise separable convolutions that are made up of depth-wise and pointwise convolutional layers. This gives the MobileNet the speed boost required and yields a much smaller network which can still compete with the larger networks on recognition accuracy.

The second component of the neural network is the multi-box SSD detector. The single shot detector network has a base of the VGG-16 neural network that is used as a feature extractor. The base is then stacked with multibox convolutional layers that perform the object detection for objects at different scales. The earlier convolutional layers of the network are wide and flat while the deeper layers have progressively increasing depths and smaller image feature maps. The earlier layers capture a large part of the image and construct simpler, more abstract features to detect larger objects while the later layers increase the feature resolution by decreasing the feature map size in order to detect smaller objects. Each of the multibox consulting layers have anchor boxes (multi boxes) of different sizes and aspect ratios for greater flexibility. During training, the ground truth bounding boxes are matched with these anchor boxes and the intersection over union (IoU) is used to calculate the fitness for optimization.



The fully connected layers of the MobileNet are removed and the single shot detectors are stacked on top of it to create an end to end object detection pipeline that works very efficiently by performing the object detection in a single pass.



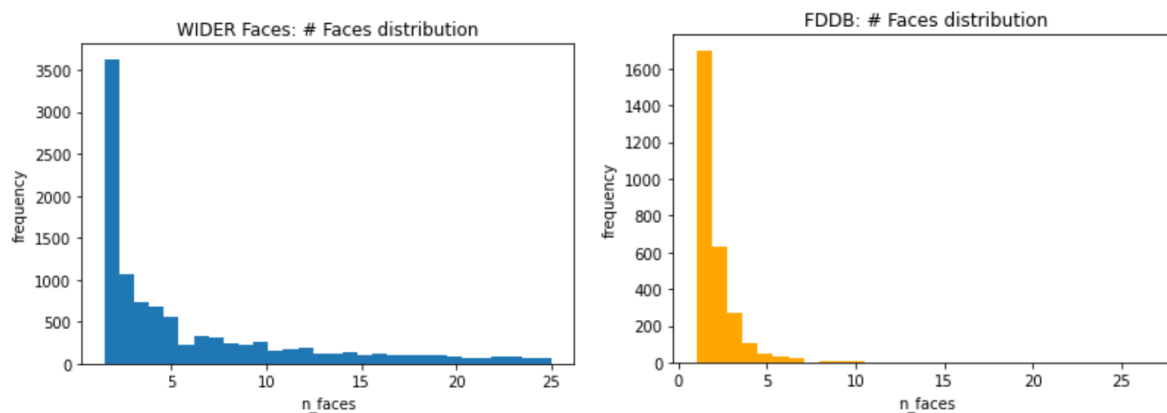
The network works very well and the performance is comparable to larger models with an order of magnitude greater number of parameters. The network does struggle with smaller objects and images with a large number of objects. This, however, should not be a problem for our application since it constitutes a single object (human face) that comprises a significant proportion of the images. Compared to non-deep learning such as Viola Jones or the HOG detector, this network can manage much larger perspective variations, distortion and partial faces which, while not necessary for our specific application, are still nice to have. The actual performance on a raspberry pi still remains to be seen but it is expected to be comparable to that of the YoloV3 tiny. The detailed analysis of the MobileNet SSD networks can be found in the experimental section of this chapter.

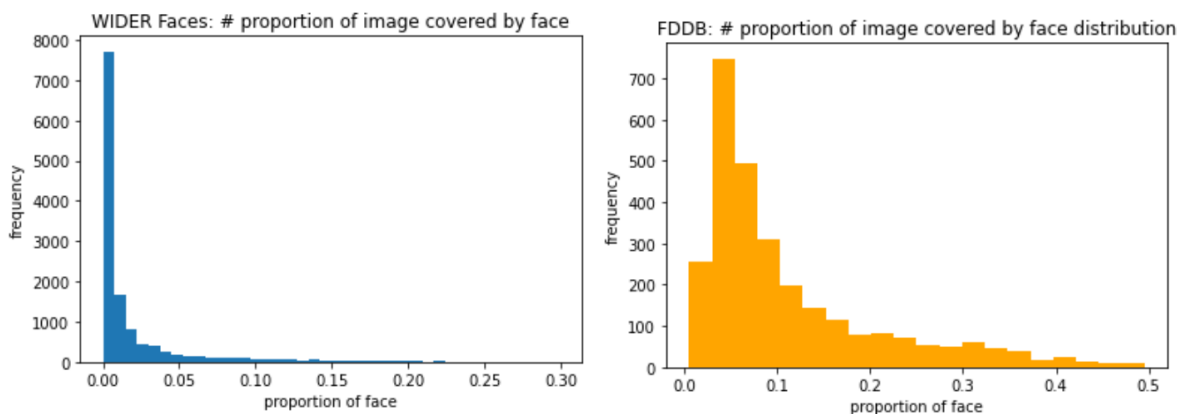
3.4 Experimental Results

This section pertains to the experiments done and the evaluations carried out to determine the best face detection algorithm for our face recognition pipeline.

3.4.1 Methodology

As mentioned above, the Wider faces dataset and the Fddb dataset was used to make a final evaluation. The Wider face dataset contains about 12,000 images across multiple categories while the Fddb dataset contains about 2700 annotated images. The most important thing that sets the two datasets apart is the average number of faces and perspectives. The Wider face dataset has a much larger number of faces and consequently, the average face is very small in proportion to the image. The Fddb in comparison has few faces per image with each face covering a much larger percentage. While the Fddb dataset with fewer images covering a larger proportion of the image is more useful to our application, I decided to test both dataset for greater variety in the testing data and to perform a more thorough analysis. The distributions of the two datasets can be seen in the following graphs:





1500 images were sampled randomly from each of the datasets and combined to form the final dataset that was used for the experimentation. For the Wider face dataset, a roughly equal number of images were chosen in the different categories.

3.4.2 Results

Pretrained model weights of each of the four algorithms were found and model interfaces were written in the python programming language to run the tests. For each of the images, the predictions from the respective models were made. Some code was written to parse the predictions in the form of bounding boxes. As mentioned in section 2, the intersection over union (IoU) with a minimum threshold of 0.4 was used to match the prediction bounding boxes with the ground truth boxes. Then, metrics such as the accuracy, precision, recall and the average IoU score were calculated. An overview of the results are as follows:

	accuracy	precision	recall	iou	time_taken
algorithm					
VJ	0.861884	0.829074	0.861884	0.641038	0.221248
HOG	0.865380	0.990014	0.795380	0.581356	0.184023
YOLO	0.911946	0.974986	0.911946	0.858066	0.485145
SSD	0.882421	0.966592	0.882421	0.736629	0.432535

The initial results show an overview of what can be expected from each of the algorithms. The YoloV3 Tiny scores the best in the accuracy metric (percentage of actual faces detected) by a small, somewhat insignificant margin. However, this definition of accuracy does not take in account the number of false detections. For that, the precision and recall metrics can be checked. The precision metric is proportional to the number of true positives detected by the algorithm. A high precision means that the algorithm returns mostly correct detections while avoiding false detections. Surprisingly, the Hog detector has a higher precision compared to more advanced deep learning detectors such as the Yolo v3 and the SSD. The margin is very small and the deep learning networks fare very well.

While the Hog detector may look like the best candidate, the recall metric gives away the reason behind the high precision. The recall metric is inversely proportional to the number of false negatives detected and higher recall means that the algorithm is detecting most actual instances (regardless of the number of false detection). The hog detector has a significantly lower recall than the rest which suggests that the algorithm is very conservative with the detections and offers face detection a marginally higher true positive rate at the cost of a significantly higher false negative rate. While the face recognition pipeline is safe from misleading information from incorrect detections, it might miss an occasional frame leading to higher overall detection time. Overall, yolo has the best compromise between precision and recall (false positives and false negatives). On the contrary the Viola Jones detector takes the opposite approach. It has a recall which is comparable to that of the neural networks which suggests that it detects the faces correctly at the same rate. It, however, has a significantly lower precision suggesting a higher number of false detection. This is troubling as this means that the system gives a lot of misleading information that can compromise the entire pipeline.

The analysis intersection over union metric is a tricky subject to deal with. While the yolo significantly outperforms every other algorithm, it is important to dig a little deeper and understand the difference between different face annotations. The yolo and the mobilenet are trained on the dataset that use face annotations that are similar to the Wider face and the FDDB dataset as both face annotations include the top of the head and the chin of the person. On the other hand, the viola jone and the hog detector were trained in a different manner and prefer to annotate on the

face only. This means that, while the IoU scores differ significantly, the algorithms may perform at a similar level. A few examples of the detections made by each algorithm were checked to confirm this hypothesis. The hypothesis is confirmed to be true so the performance on the average IoU metric is considered to be a tie.



Ground Truth



Viola Jones



Hog Detector



YOLO V3 Tiny



MobileNet SSD

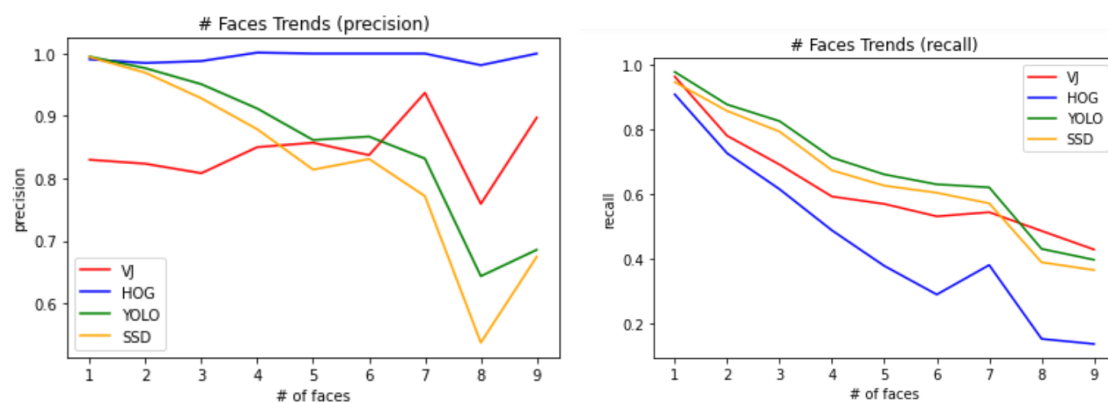
The Yolo V3 tiny model comes out on top according to most of the accuracy metrics. However, the detection time reveals the major flaw for deep learning systems. The Yolo V3 tiny has the highest detection time out of all of the algorithms with the MobileNet SSD not much far behind. In comparison, the Viola Jones algorithm and the Hog detector have an up to three times lower detection time. This means that Viola Jones or a Hog detector can run on a raspberry pi at 6 fps compared to less than 2 fps for the Yolo V3 Tiny.

The overall results of the experiments make either the Yolo or the Hog detector the front runner depending on the criteria. In order to perform a more thorough

analysis, it may be worth studying the detection metrics for each of the algorithms with reference to the type of image being processed.

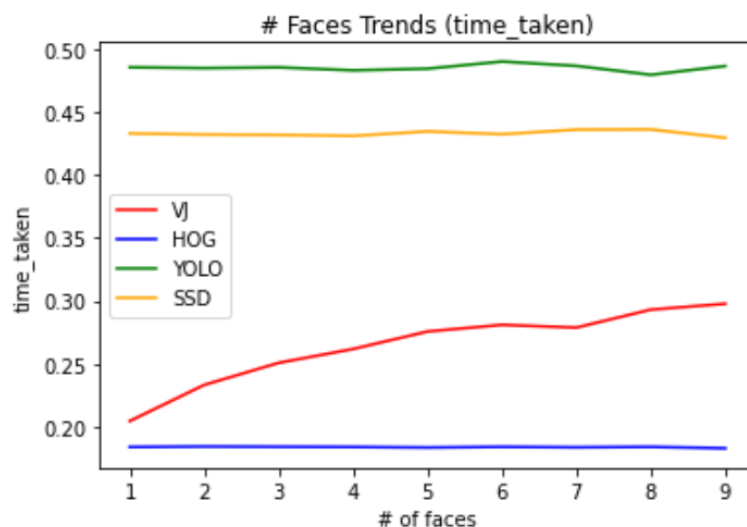
3.4.3 Number of Faces

The datasets tested have different types of images and one of the main themes along which the images differ is the amount of faces in an image. It is interesting to study the effects of the number of faces in each of the algorithms in order to determine the best suited choice. The dataset was filtered out to remove outliers and images with too many faces since our application involves face detection of only one person at a given time. Only the images with 10 or less faces were considered and the dataset was further divided in 10 slices (1 faces, 2 faces, 3 faces...) and the general experiment mentioned above was rerun. The performance metrics for each slice were calculated individually and compared with each other in order to study the trends.



The graphs above show the trend of the precision and recall metrics as the number of faces in the image rises. The precision of the Viola Jones and the Hog detector remains the same which might suggest the algorithms fare better than the deep learning networks. However, further analysis reveals that this is not the case as the traditional algorithms simply stop detecting any faces in images with a large number of faces causing both low true and low false positive rates leading to higher precision. This is further verified in the recall graphs. Unsurprisingly, the

The recall trends toward at a similar rate. The deep learning networks maintain their recall very well while the traditional algorithms do not. The worst offender is the hog detector for which, the recall nosedives (while maintaining a high precision) confirming the conservativeness of the algorithm. That said, The hog detector works very well for images with a small number of faces which is the primary goal of the project undertaken in this dissertation.

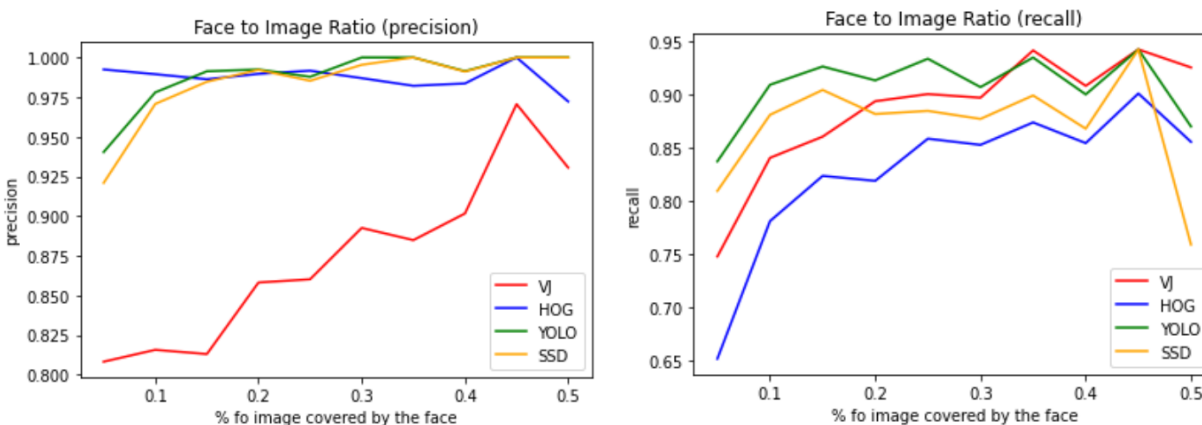


The graph above shows the detection time of each of the algorithms on a raspberry pi. For most of the algorithms, the detection time is constant. The only exception is the Viola Jones for which the detection time rises with the number of faces. The Hog detector remains the fastest algorithm for each dataset slice.

3.4.4 Face to Image Ratio

Apart from the number of faces, the size of the individual faces should also be considered. The dataset contains images in various situations with faces of all shapes and sizes. We are primarily interested in the images that contain people looking directly in the camera for whom the faces cover a large portion of the image. This subsection pertains to the testing done in order to evaluate the performance of the algorithms on faces of different sizes. To accomplish this, the face ratios were calculated by dividing the face areas by the total image size. For example, a ratio of 0.25 means that the face covers 25% of the total image area. The dataset was divided in 10 slices (0 - 0.05 , 0.05 - 0.1 ...) and the general

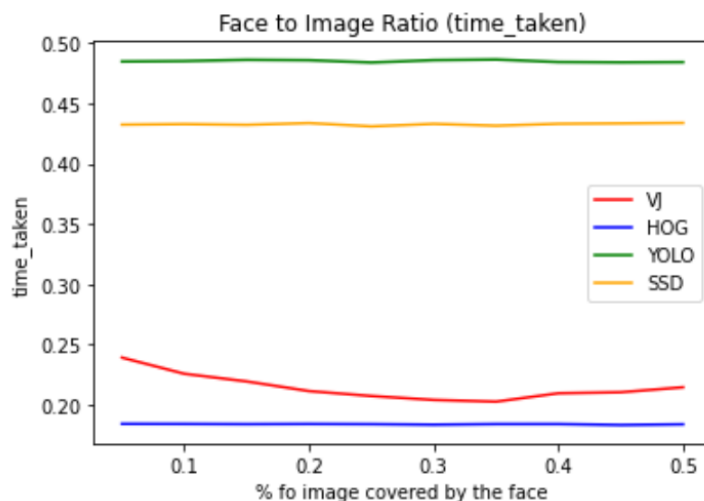
experiment explained above was rerun. The performance metrics for each slice were calculated individually and compared with each other in order to study the trends.



The precision and recall increase for all algorithms as the face to image ratio rises. This is expected as it is easier to identify faces that are more prominent and cover a larger area. Furthermore, the image dimensions are standardized to 512x512 which means the faces with a higher ratio are higher resolution as well making them easier to detect.

According to the precision and recall metrics, the algorithms behave in line with in observations in the previous experiments. The Viola Jones and the Hog detector perform comparably to the neural networks on images with a high face to image ratio. However, the performance of the traditional algorithms drops significantly as the ratio decreases while the neural networks maintain performance reasonably well. The drop in performance of the viola jones is significantly worse for the precision metric all across the board confirming the excessive generosity of the algorithm that causes an increasing amount of false detection as the ratio drops. Similarly, the Hog detector is the worst performer in the recall department with the recall significantly dropping as the ratio drops. This reiterates the excessive conservativeness of the Hog detector that forgoes true detection in order to minimize false detections. That said, the Hog detector performs reasonably well on faces that cover a large portion of the image. The benefits of the deep learning

networks are not observable unless the faces cover a much smaller portion of the image which is largely irrelevant to the practical problem at hand.



The detection time remains constant across the board for all of the algorithms. The detection time of the viola jones increases slightly for very small faces but this can easily be attributed to the fact that images with smaller faces typically contain a large number of faces which increase the complexity and slows down the detection process of the Viola Jones algorithm.

3.4.5 Conclusion

The Viola Jones algorithm can be ruled out at the very beginning. It has a very large false positive rate which is arguably worse than false negatives. All it offers is marginally higher detection rate on images with constrained backgrounds and a single face that covers a large portion of the image at the cost of a much lower performance on more complex images and a higher false positive rate all across the board. It is also computationally comparable to the hog detector which leaves no reason to pick it over any of the other algorithms. Similarly, the MobileNet single shot detector can also be ruled out as it offers no unique benefits and Yolov3 outperforms the MobileNet in every accuracy metric. All the MobileNet offers is a slight advantage in the detection time but is insignificant and can be written off.

Hence, the final selection of the face detection algorithm is a choice between the Hog detector or the Yolo V3 Tiny depending on the final objective.

From a purely performance point of view, the yolo algorithm outperforms the rest. However the margin of victory is not as significant as one might hope. Moreover, the yolo algorithm is the most computationally heavy of them all and runs three times slower than the hog detector which has a comparable performance. In addition to this, the main issue with the hog detector is missed detection which can be easily remedied by running more video frames for each face detection attempt which we can as we have to run the face recognition pipeline just once per scan and a recognition time of up to a few seconds is still acceptable. The multiple frame runs for a hog detector are easily compensated (and then some) by the three times higher frame rate. Furthermore, the access control system has to mostly deal with frontal face views and images with a small number faces that are detected reasonably well with a Hog detectors. Most of the advantages of the deep learning neural networks are observed only on more complex images with a large number of difficult to detect faces.

Lastly, I am reluctant to use a Yolo algorithm on the raspberry pi due to the high detection time as my tests indicated a frame rate of less than 2 fps leading to a very high opportunity cost of processing blurry frames or frames with bad poses. The hog detector holds out very well for a non-deep learning algorithm and only has a slightly lower performance in exchange for a 300% boost in the detection speed and subsequent frames per second. Therefore, I am inclined to choose the Hog detector as the face detector of the application.

Section 4 - Face Alignment

Successful face detection results in bounding boxes that strictly content the faces of the humans. The next challenge is to conform the detected faces to a common standard before the rest of the face recognition pipeline can be executed. This intermediate stage between face detection and face verification or identification is known as face alignment. Face alignment can be considered as a form of data normalization or zero centering and it is considered useful because it provides a layer of standardization that allows for more robust face recognition as the overall pipeline is less likely to be affected by variables like human pose, facial expressions, face resolutions and orientation. For this reason, it is recommended by many popular face recognition algorithms such as eigenfaces, fisherfaces and local binary patterns (lbp) of faces. After a lot of research and experimentation, it is generally accepted that a higher accuracy of face recognition can be achieved by performing face alignment.

In general, the face alignment algorithm consists of two phases. The first phase deals with the identification of the geometric structure of the detected faces by detecting the location of various facial landmarks. This is a classic object detection problem and has traditionally been solved with machine learning algorithms and computer vision techniques. Like other subfields of computer vision, face landmark detection is also rapidly moving towards deep learning and most of the recent research and the state of art algorithms are based on convolutional neural networks that offer higher performance with little preprocessing. That said, landmark detection and face alignment is extremely time sensitive as it is typically expected to consume only a small percentage of the time taken by the pipeline. For this reason, traditional algorithms are more commonly used compared to detection or recognition due to the speed boost.

The second phase utilizes the face landmarks to align the face by warping and transforming the image to conform it to a standard. More sophisticated techniques involve imposing a predefined model that warps the entire perspective of the image to center the face. This leads to maximum conformity as it is guaranteed that the face landmarks of two warped faces would overlap perfectly. Despite the higher

degree of standardization, this approach distorts the face and may lead to information loss. The second, more simplistic approach is linear transformation that attempts to canonically align the face based on normalized image translation, rescaling and rotation. This is desirable as it preserves the structure of the face while providing an acceptable degree of conformity. The basic goals of such alignment are as follows:

1. The faces should be centered in the image.
2. The faces should be rotated such that the eyes are horizontally aligned with each other (the y-coordinate of both eyes is identical).
3. The faces should be scaled to an identical size and resolution.

Despite its simplicity, this approach works very well and is recommended. Furthermore, most of the face identification algorithms that are discussed later are trained on faces where facial structure is preserved (which is the latter approach) or no preprocessing is done leaving us with little choice but to use the second approach of centering, rotation and rescaling faces instead of a 3D perspective transformation.

4.1 Performance Evaluation

Performance evaluation for face alignment is tricky due to it being an intermediate stage with no clear result that can be quantified or evaluated. The performance and the suitability of an algorithm can be evaluated by proxy by evaluating the performance of principle constituent algorithm, landmark detection since the subsequent alignment algorithms are strictly deterministic and depend solely on the quality of the face landmarks detected by the landmark detector. The face landmark detection problem statement is as follows. Given an image and a list of bounding boxes that represent each detected face, for each of the detecting faces, predict the location of a list of an arbitrary number of predefined features such as eyes, nose etc and return a fixed length list of spatial coordinates for each of the features. Mathematically speaking, this boils down to a regression problem as the

output represents the actual real world value. Hence, metrics normally used for regression analysis can be used for performance evaluation.

The first metric that can be used to evaluate the fitness of the landmark prediction is the mean distance between the predictions and the ground truth. For a single face this would be the sum of the euclidean distances between the coordinates that represent the predicted location and the actual location of each of the landmarks. To obtain a single fitness value for the entire dataset, the mean square error (MSE) will be used.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2$$

The mean squared error works great for relative comparison between two predictions but is incomprehensible to humans for absolute comparisons. Therefore, for a single face, the mean percentage error will be used to establish a minimum viable performance and evaluate the algorithms on its own in absolute terms. In addition to the mean percentage error, the maximum and minimum percentage error in the landmarks will be calculated for the landmarks of individual faces in order to examine the extremes and outliers to establish upper and lower bounds on the performance. A perfect face landmark detector, for example, would have an mean squared error of zero which means that the euclidean distance between predicted landmarks and actual landmarks is zero. More obviously, the mean, maximum and minimum percentage errors would also be zero which means that the predicted face landmark coordinates are identical to the ground truth.

The final metric that was considered is the failure rate, or the proportion of the faces, for which, no landmarks were detected. This is a binary metric that can be analysed by calculating the precision, recall and the accuracy. However, a few of the most popular algorithms used for landmark detection such as the Ensemble of Randomized Trees (ERT) model have an a priori assumption of the existence of the face in the given bounding box. This means that the landmarks are always estimated regardless of the actual existence of a face. Therefore, the failure rate is always zero and the precision and recall metrics lose their meaning which means

that no meaningful comparison between algorithms can be made using the failure rate.

4.2 Datasets

The experimental tests designed to evaluate, compare and choose the ideal landmark detection algorithm were undertaken using the 300W face landmark dataset. The 300W dataset is made up of a collection of various face recognition datasets such as the Labelled Faces in the Wild (LFW), Annotated Faces in the Wild (AFW) and the IBUG dataset which makes it the ideal candidate as it makes experimentation with multiple datasets redundant. The images are annotated manually using 68 separate face landmarks as shown below.



The dataset contains 300 indoor and 300 outdoor images and were carefully selected to represent a characteristic sample of face in fully unconstrained condition. Hence, there is a huge variation in the types of images as the dataset contains images with a wide variety of lighting conditions, distance from camera, expressions, identity, human body pose and visibility as well as images with faces that are partially obstructed. A closer inspection reveals that the images focus on the humans instead of the background as 50% of the images contain a single face with a further 10-15% of the images containing only two faces. The median face is

considerably large and visible with a median face resolution of 292x292 which is expected due to the difficulty in the identification and annotation of face landmarks for very small faces. That said, the diverse set of images is definitely at least as challenging as the real world office environment. This means we can be assured that an algorithm that works well on the dataset will also work well for the face alignment component of the proposed access control system.

4.3 Algorithms

Landmark detection is a popular area of research and has many applications out of face alignment in face recognition such as face swapping, gaze detection, drowsiness detection and augmented reality. Popular modern day applications such as mobile camera filters are built on landmark detection which generates a lot of research and business interest in the field. This has led to the development of various algorithms.

The earliest landmark algorithms were statistical in nature and were based on a face mesh that fit on detected faces. The generated meshes were then used to estimate the location of the landmarks. Examples of such algorithms include the Active Shape Model (ASM) and the Constrained Local Model (CLM). These algorithms kickstarted the field of landmark detection and served as the proof of concept in the industry. While interesting, the algorithms were not powerful and complex enough to work in real world scenarios such as employee access control in office buildings. With the advent of machine learning, new face detection algorithms were developed that use automated feature extraction and trainable models that led to a drastic increase in performance making the second generation suitable for real work applications. Examples of the second category include tree based ensemble algorithms such as the Ensemble of Randomized Trees (ERT) algorithm that is described, tested and evaluated in detail in the following subsection.

Today, just like the vast majority of predictive algorithms families, the state of the art algorithms in facial landmark detection algorithms are deep learning based

neural algorithms. These latest neural networks offer the most accurate predictions with the lowest prediction error and work just as well in more difficult situations. In fact, these new networks have started to eclipse actual humans in face landmark detection with the only significant drawback being the hardware requirements and detection times. The most popular deep learning based landmark detector is the Multi task Cascaded Convolutional Neural Network (MTCNN). The MTCNN is the second algorithm that is tested in the experiments and evaluated.

4.3.1 Ensemble Randomized Trees (ERT)

Arguably the most popular algorithm used for face landmark detection is an ensemble of randomized trees (ERT) proposed by Vahid Kazemi and Josephine Sullivan in 2014 in the paper “One Millisecond Face Alignment with an Ensemble of Regression Trees”. The algorithm comprises a cascade of gradient boosting trees that generate an ERT face template and iteratively refine it to predict the locations of the landmarks. Even though there are more complex, deep learning based algorithms available today, the ERT algorithm is still widely used due to excellent public implementation, very high precision and detection time (hence the “One Millisecond” in the paper title).

The ERT landmark detection algorithm starts off with a generic face shape and utilizes a cascade of regressors to iteratively improve the shape and predict the landmarks very efficiently in near real time. The initial shape is usually the mean shape in the datasets and the regressors make their interactive prediction based on image features such as the pixel intensity values. The cascade of regressors are trained jointly. With each training iteration, the prediction accuracy rises and the training process is repeated until sufficiently high accuracy is achieved. The individual regressors are tree based and use the gradient boosting algorithm to fit to the target vector.

The implementation of the ERT algorithm used for further testing is from the open source computer vision python library called DLib. The algorithm provides a very high prediction accuracy and fits the landmark features very well. The base

template and the interactive improvement dramatically speeds up the prediction process by greatly simplifying the problem and removing the binary classification phase altogether. Further information on the detection time can be found in the testing subsection, but casual testing demonstrates almost instantaneous prediction explaining the popularity of the algorithm. That said, this approach has a drawback as it strictly assumes the existence of a face in the provided bounding box. The iterative improvement process leaves no space for a negative classification and the algorithm will approximate the landmarks even in the absence of an actual face without providing a confidence score. In other words, the robustness of the algorithm significantly depends on the preceding face detection phase. An incorrect or misaligned face will mislead the algorithm without warning,

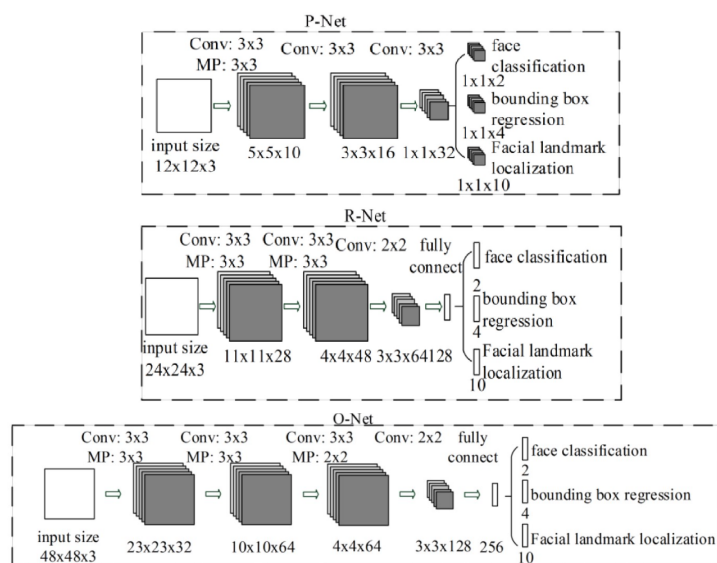
Fortunately, good results were observed in the face detection section and the selected face detector specifically prioritizes the reduction of false positives and demonstrates precision detection. This significantly reduces the impact of drawbacks of the ERT algorithm making it a promising candidate for face alignment in the face recognition pipeline

4.3.2 Multi task Cascaded Convolutional Neural Network

The MTCNN is an ensemble of convolutional neural networks that combines the face detection and the landmark detection and makes a joint prediction. Unlike the 68 landmark industry standard, it is designed to predict only five landmarks (left and right eyes, nose and left and right corners of the mouth). The MTCNN comprises a preprocessing component followed by a cascade of three separate neural networks, the proposal network (PNet), the refine network (RNet) and finally the output network (ONet). The main job of the preprocessing component is to rescale the input image to various sizes in order to generate a pyramid of images. Subsequently, this pyramid is given as input to the three network cascades for joint face and landmark detection.

The first state of the cascaded network, the proposal network, is a shallow fully convolutional neural network (FCN) designed to rapidly generate candidate

windows. The PNet predicts the initial candidates for face detection and landmark detection and uses bounding box regression to predict the bounding boxes. Standard post processing is done to combine overlapping regions and the candidate windows are passed on to the second network. The refine net is a deeper, more complex neural network with a fully connected layer. This aim of the refine net is to reduce the number of candidate windows, calibrate the results of the regression and further combine the overlapping bounding boxes. The RNet has a fixed input at 24×24 and a fixed length output. It outputs a 15 length vector that constitutes a binary output that represents whether the input region is a face, a four length vector that represents the bounding box for the face and a 10 length vector that represents the x and y coordinates of each of the five landmarks. The output vectors are then passed on to the third network. The aim of the ONet is to further refine the predictions and add more details to the face. The structure is similar to the RNet as it is a convolutional network with four convolutional layers and one fully connected layer for the predictions. Like the RNet, the input is fixed at 48×48 and the the output is a similar 15 length vector that represents the binary classification results, bounding box for the face and the coordinates of the facial landmarks. The face landmarks can then be used for the face alignment by calculating the required rotation angle, scaling ratio and the center of the face.



One of the main advantages of the MTCNN is the joint prediction of the face classification, localization and the landmark detection as it reduces the overall complexity of the face recognition pipeline. The quality of the prediction is also

very good and the algorithm demonstrated impressive performance on mainstream face landmark datasets even in partially obstructed faces with difficult to distinguish features. However, it does so at the expense of time as the precision takes precedence over detection speed for state of the art performance. This leaves some doubt over whether the MTCNN would be appropriate for our application since the results of the landmark detections are only to be used indirectly for alignment. Experimental testing was carried out to compare the MTCNN with the ERT algorithm to resolve the doubt and select the best choice in context of the real world application.

4.4 Experimental Results

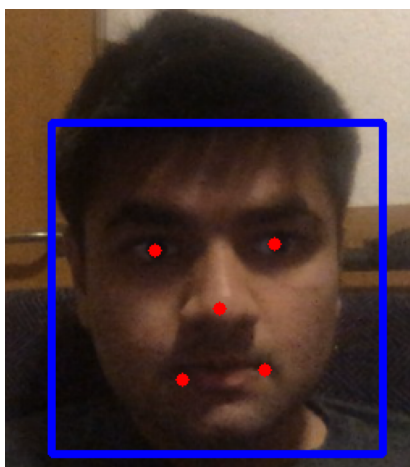
This section pertains to the experiments done and the evaluations carried out to determine the best face landmark detection algorithm that can be used to face alignment between the actual recognition. As mentioned above, the remainder of the alignment process is not tested due to the fact that it is an intermediate stage with no clear fitness function.

4.4.1 Methodology

First, pre-trained models for both the Ensemble Randomized Trees (ERT) model and the multi-task Cascaded Convolutional Networks (MTCNN) and modular programming interfaces were implemented in the python programming language to use the models. For the regression trees model, the `68_face_landmark_detector` from the `dlib` python library was used while a third party implementation of the `mtcnn` was found on github along with pretrained weights for the `mtcnn`.

The images 300W dataset was used to evaluate either landmark detection algorithm and then select the most appropriate choice. There is a difference between the number of the landmarks detected by each algorithm and the number of landmarks found in the annotations of the dataset. The ERT algorithm predicts

68 landmarks which correspond exactly with the 68 landmarks that are annotated in the dataset. However, the MTCNN only predicts 5 landmarks, of which, two do not correspond perfectly with any of the 68 landmarks predicted by the regression trees and the dataset. To remedy the situation, the nose, left mouth and right mouth landmarks are filtered directly from the original 68 and landmarks for the center of the left eye and the center of the right eye were calculated by extrapolating the landmarks that correspond to the the outlines of each eye. The five resultant landmarks correspond perfectly with the 5 landmark format of the MTCNN and the rest of the face landmarks are discarded. The utilized landmarks are illustrated in the image below.



At this point, the chosen face detector (the Hog detector) is used to detect faces in each of the images in the dataset. For successful face detections, the landmark detectors are used to predict the landmarks. Since the images in the dataset have landmark annotation for only one face (out of possibly multiple faces), the overlap was checked. Only the face detection with an overlap with all 68 original annotations was considered and the rest were discarded. For each face landmark prediction on an eligible face, the time taken for detection, the sum of squared errors, the mean, the minimum and the maximum percentage error (euclidean distance) was calculated. The metrics for each face were saved in a csv file and analyzed. The individual metrics were grouped by the algorithm to obtain singular metrics for each of the algorithms.

4.4.2 Results

Before discussing the algorithms, the exact goal should be determined in order to determine what constitutes “good enough” for face landmark detection. Face alignment is considered a secondary component of the face recognition pipeline and it is absolutely vital to complete it in the shortest time possible. Furthermore, the accuracy required for alignment is not extremely high as only the general structure or pose of the face needs to be understood and small detection errors in individual landmarks are of little consequence. As long as a detection is accurate enough to calculate the center of the face, the required rotation angle and the rescaling ratio, the detection can be presumed to be “good enough”. Hence, the speed of the algorithm takes precedence over the precision as long as the accuracy is somewhat reasonable. The initial results of the experiment are as follows:

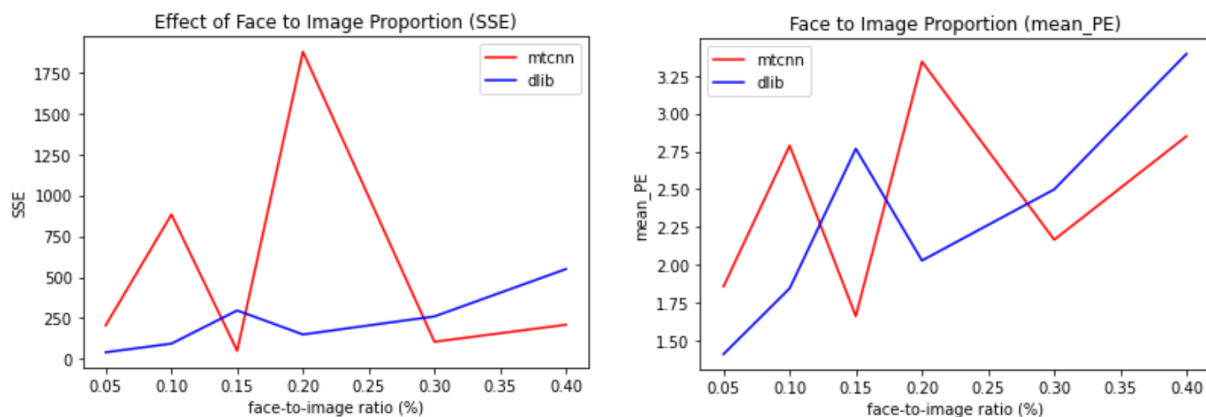
	time_taken	SSE	mean_PE	max_PE	min_PE
HOG + dlib	0.239947	109.778583	1.783193	3.156021	0.753332
MTCNN	1.749722	469.443658	2.503717	3.951898	1.316449

Both algorithms perform very well and the sum of square error is extremely low for either one of them. For reference, a SSE of 109.77 means that the predicted landmarks are, on average, just 5 pixels away from the ground truth which, for all intents and purposes, is negligible. This is further confirmed by the percentage error metrics. The average percentage errors for the Dlib and MTCNN detector are just 1.7% or 2.5% retrospectively. Even the maximum percentage error, which is the worst of the five landmark predictions, is just 3.15% and 3.95% respectively for the two detectors which is well within the acceptable range. Overall, the difference in the performance between the two algorithms according to the accuracy metrics is insignificant.

The detection time is a completely different story. As mentioned above, the MTCNN is a convolutional neural network that combines the face detection and the landmark detector. Therefore, the detection time comparison was done with the MTCNN and the combined detection time of the Hogface detector and the Dlib

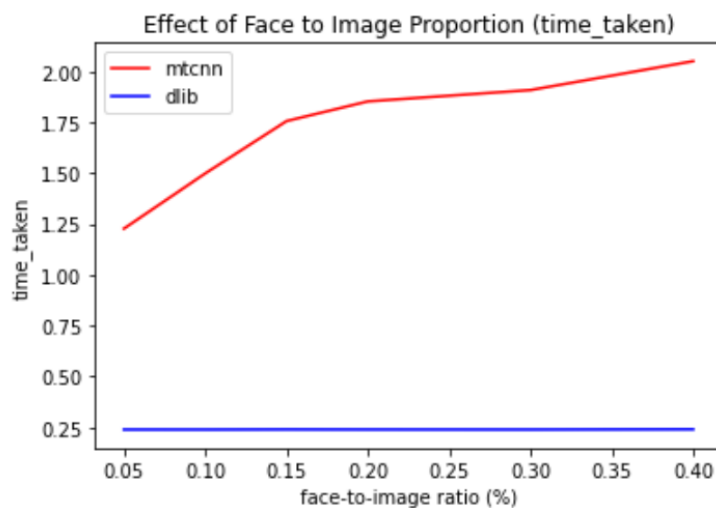
landmark detector. The latter combination widely outperforms the deep learning algorithm by a long shot. The ERT algorithm, surprisingly, holds true to its name as the landmark detection takes a mere 0.02 seconds on a weak raspberry pi. The detection time for the hog face detector and dlib landmark detector combination is mere 0.23 seconds compared to 1.8 seconds for the mtcnn. Not only is the mtcnn is eight times slower than the traditional approach, it is also too slow from an objective, absolute point of view to be run in real time. With no discernible difference in performance and the overwhelming speed advantage, the dlib detector is an obvious choice for face alignment with absolutely no reason to choose the MTCNN except for images with unique perspectives and a generous amount of time to process them.

Before concluding the selection process for sure, a closer look was taken. One thing to consider is the face to image ratio. Convolutional neural networks work very well in obscure situations and it is useful to ensure that the performance of the traditional approach does not drop as the face to image ratio decreases and the complexity of the problem rises. To do this, the face to image ratios were calculated and the dataset segmented in 10 slices according to the face ratio (0 - 0.1, 0.1 - 0.2 ...) and the general experiment mentioned above was rerun. The performance metrics for each slice were calculated individually and compared with each other in order to study the trends. The graphs representing the trends in performance as the the face to image ratio varies are as follows:



Surprisingly, the error metrics rise as the face to image ratio rises. Upon further inspection, this can be explained simply by the higher resolution of the face that is

caused by a larger face. The metrics are based on the euclidean distance is linearly proportional to the resolution of the face. A mistake of the same magnitude would yield higher error metrics in faces that are composed of more pixels, A visual inspection confirms this hypothesis as the landmarks appear to fit better in images with bigger faces despite the higher error metrics. That said, the two algorithms demonstrate similar performance in terms of accuracy for all of the segments in line with our expectations and the original results.



The detection time for both algorithms increases steadily as the face to image ratio increases which is to be expected due to the higher number of pixels to process. Unsurprisingly, the overwhelming difference in the detection time observed previously holds true of all situations. The ERT algorithm is an order of magnitude faster in all situations, hence, confirming its original selection.

Section 4 - Face Recognition

Face recognition is the third and final step in the end-to-end pipeline. It refers to the actual mechanism that is used to compare faces. Face recognition is generally treated as two distinct problems, namely, face verification and face identification.

Face verification is a 1:1 comparison problem and deals with the comparison of two faces in order to determine the similarity. It is a binary classification problem in which the output is either “yes” if the faces belong to the same human or “no” if they belong to different people. Face verification is carried out by comparing the “distance” between the respective numerical embeddings (mathematical representation vectors) of the two faces and assigning the pair a verification score. A threshold that is determined through trial and error is used to make the final judgement. The verification is considered successful if the verification score (the distance) is higher than the threshold and a failure otherwise.

On the other hand, face identification is the generalized form of face verification. It refers to the problem of recognizing the face of a human from a predefined collection of human faces. The outputs of face recognition are two folds. The first goal is to determine if the face in question is recognized (that is, matches one of the faces in the collection) or is unknown. The second goal is the determination of the person, if any, the face belongs to. This is typically the last component of the pipeline and does not require complex algorithms of its own. It is typically carried out by verifying the new face with every face in the database and checking the verification score. If the verification score of the highest performing face is above a certain threshold, the recognition is deemed successful and the identity of the person is returned as output. The recognition is deemed a failure and the face is marked as unrecognized if the highest score is below the threshold.

Despite their different natures, both subcategories use the same underlying comparison mechanism. A mathematical representation called a face embedding is calculated using an algorithm. This mathematical representation can be used in arithmetic operations and is used to calculate euclidean distances to achieve a

single face similarity score. As mentioned above, this score is compared against a predetermined threshold to convert the similarity score into a positive or negative verification. In case of face recognition, the face with the lowest distance is selected and compared with the threshold.

4.1 Performance Evaluation

The main goal of the algorithm evaluation is to determine how well the mathematical embeddings represent the original face. Since each of the algorithms produces embeddings of different dimension and scale, only relativistic comparisons can be evaluated which is why the algorithms are evaluated on their final discrete prediction rather than the actual distances between the corresponding face embeddings.

Face verification is a standard binary classification problem, therefore, standard classification metrics such as accuracy, precision and recall are used to determine the performance of the algorithms. In addition to these measures, the number of false positives and false negatives is also analyzed in order to gain a deeper understanding of the behaviour of the algorithms. In case of the face verification, the following definitions of the metrics are used:

True Positive: Algorithm approves verification for images of the same person;

True Negative: Algorithm rejects verification for images of different people;

False Positive: Algorithm approves verification for images of different people;

False Negative: Algorithm rejects verification for images of the same person;

Face identification introduces more complexity but can still be treated as a binary classification problem by changing the definitions. Therefore, the usual metrics such as accuracy, precision and recall can be used for evaluation. Due to the 1:N nature of the problem, there false negatives and false positives can be further divided into the misclassifications and misrefusal metrics. In case of the face recognition, the following definition of the metrics are used:

True Positive: Algorithm correctly selects image of the same person from dataset;

True Negative: Algorithm rejects identification when the selected person is not present in the dataset;

False Positive: Algorithm selects a person when the selected person is not present in the dataset;

False Negative: Algorithm rejects verification when the selected person is present in the dataset or selects wrong person;

4.2 Datasets

Tests and experiments have to be done in order to evaluate and compare the performance of the face recognition algorithms. These tests are carried out on several publically available datasets. It is important to select the appropriate datasets that reflect the settings of the final system as closely as possible. As mentioned above, the consumer system designed and implemented in this dissertation is supposed to be used for access control and employee verification inside office buildings. That said, the datasets are selected to represent a diverse set of conditions in order to simulate and indirectly test different backgrounds and scenarios. The following subsections deal with the several face recognition datasets that are used to run the experiments:

4.2.1 Labeled Faces in the Wilds (LFW)

The first dataset selected for use in the experimentation is the Labeled Faces in the Wild (LFW) dataset. The LFW datasets were made public in 2007 and are widely considered as the main reference benchmark for face recognition algorithms. It contains 250x250 dimensional images of about 5800 different people. The total number of images is around 13,000 which suggests an average of 2.2 images per person.

The images in the LFW datasets are relatively unconstrained and seem to represent the real world making it an effective benchmark for real world performance.

However, most of the images are frontal views that are easy to distinguish and identify which means the real world performance may be a bit lower than what the dataset might represent. That said, the images contain a wide variety of illuminations, backgrounds and poses and the popularity of the dataset has resulted in the creation of various supplementary datasets. A few of these third party datasets are used to add information such as age, gender and race to the images which allows for a more comprehensive analysis. The human accuracy for the LFW dataset is 97.2% in 1:1 face verification tasks. This human performance is used as a benchmark to quantify the performance of the algorithms in absolute terms.

4.2.2 Cross Age Celebrity Dataset (CACD)

The second dataset used in the experimentation is the Cross Age Celebrity Dataset (CACD). This dataset was scraped from the popular movie rating website, IMDB, and contains roughly 160,000 images from different life phases of 2,000 different celebrities. As suggested by the name of the dataset, the focus is on the diversity in ages of the same person in order to test the resilience and robustness of the face recognition algorithms. The ages range from 16 to 62 and the dataset contains a much larger number of images per person compared to the LFW dataset which allows for a larger number of distinct positive comparison pairs.

The CACD dataset serves as the more complicated dataset and manual inspection reveals that it is definitely harder than normal real world conditions. In addition to the larger age differences, the CACD dataset contains more difficult to recognize facial poses. Therefore, the final real world performance of an algorithm can be expected to lie between the performances of the LFW dataset and the CACD dataset. For reference, the human accuracy for the LFW dataset is 87% in 1:1 face verification tasks reflecting the significant increase in difficulty. The differences in the performance between the CACD and the LFW dataset are discussed from various perspectives in the following sections of the dissertation.

4.3 Algorithms

Face recognition (verification of identification) is arguably the most important component of the pipeline and forms the crux of the problem. It solves the actual problem of comparing faces with the indirect support from the previous components of the pipeline. Due to its undeniable importance, research in face recognition far outweighs the research in face detection and alignment. On its own, this component is possibly the more important and widely researched sub-field of the broader computer vision field. Face recognition is arguably much more complex compared to the standard object detection and recognition problems as it involves relative comparison of two objects instead of simple bounding boxes and classification labels.

Research in face recognition technologies dates back to the 1970s with the majority of the earlier research focusing on statistical methods such as logistics regression. These earlier models relied on hard coded, hand crafted features that represent various facial features and face recognition remained an impractical curiosity for a long time due to the simplistic algorithms and the lack of processing power for real time face recognition. The 1990s brought the earliest practical algorithms such as the EigenFaces, FisherFaces and the LBPH algorithms. These algorithms were the first ones to propose semi-automated feature extraction using statistical techniques such as eigenvalues and automated learning. This removed the restrictions of hard coded features and drastically improved accuracy which led to real world applications of face recognition for the first time. Despite the significant improvements, the new crop of algorithms was still not good enough for unconstrained environments and generally sensitive to illumination and contrast. In addition to the sensitivity, the algorithms could only perform reasonably well on a narrow range of poses such as full frontal images and did not work well with complex backgrounds and unusual poses. Finally, these algorithms had to be trained on each dataset and performed poorly on untrained images which meant retraining the model at every new addition to the dataset making online learning exceptionally hard.

As with the other components of the pipeline as well as the broader computer vision field, the majority of the new research is based on convolutional neural networks that require little to no preprocessing and take care of everything from feature generation, feature selection and the actual classification. The latest crop of algorithms has finally made face recognition viable in practical, real world applications and is extremely resilient to variation in illumination, contrast, pose, facial expression etc. The latest convolutional neural network based face recognition has demonstrated superior performance to even humans.

Today, the research and the real world industry exclusively relies on convolutional neural networks and the earlier statistical approaches are considered relics. Furthermore, a great deal of effort has been spent to increase the efficiency of these complex networks making them somewhat viable for a raspberry pi. This is why, unlike the earlier components, only deep learning based techniques are considered and experimented with in this thesis and deep learning based techniques are the only ones that perform comparably to humans and do not require retraining at every new addition to the dataset.

Generally speaking, all of the deep learning algorithms are designed to output a fixed dimensional mathematical vector that is supposed to represent a human face in a mathematical form which allows for arithmetic operations. This representation is called a face embedding and is used to mathematically compare faces for face verification and face identification. The standard method of comparison involved the calculation of a spatial distance (euclidean distance, cosine distance) between the embeddings and using an empirically determined threshold to determine the final classification. Faces with spatial distances lower than threshold are predicted as the same person while faces with a spatial distance higher than the threshold are predicted as dissimilar people.

The five deep learning based, face embedding calculation algorithms discussed and experimented with in this dissertation are the Facenet, the MobileFaceNet, the ArcFace, the CosFace and the VGG-Face2 algorithm. A brief overview of each of these five algorithms can be studied in the next subsections.

4.3.1 Facenet / OpenFace

Facenet is widely considered as the state of the art in face recognition and was first proposed in 2015 by Google researchers. It is built on a specialized adaptation of the standard inception convolutional neural network that is used for object recognition and offers an end-to-end, unified encoding system that maps an image into mathematical vectors. As with most deep learning based techniques, the facenet model takes a cropped and aligned image of a face and outputs a 128 dimensional embedding that can be used to calculate distances and determine final prediction. Facenet does not introduce any new algorithms but rather offers a different approach to the pre existing inception network.

The most important concept in the Facenet algorithm is its loss function. Facenet introduces and uses the groundbreaking triplet loss function that is specifically designed to be used for face recognition. The triplet loss function uses an anchor image, positive (matching) image and a negative (nonmatching) image and is designed to minimize the distance between the anchor and the positive image while maximizing the distance between the anchor and the negative image. The intuition behind this concept is self explanatory. We would like the loss to be high for dissimilar pairs and low for similar pairs of images. The formal definition for the triplet loss is as follows:

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

The triplets have to be carefully selected to ensure fast convergence. Essentially, we would like to select a positive image with a large distance from the anchor and a negative image where the distance is small to ensure the triplet holds the largest amount of learnable information.

The feature extractor layers of the inception network are used to extract the features and the network is trained using the adam optimizer and ReLU activation layers. The research paper tests two pre existing models, namely the Zeiler&Fergus model and the Inception Model. The Inception model is much lighter and has as

much as 20X less parameters and is upto 5 times faster while having comparable performance. The authors also recommend minimal preprocessing which is limited to landmark detection and 2D face alignment. The overall architecture of the inception based models are as follows:



Facenet is a proprietary algorithm and the original model was not released by the authors. Therefore, an open source implementation on the same architecture called OpenFace is used for testing. The Facenet algorithm works very well and was one of the first algorithms to exceed human performance making it an ideal candidate for further testing.

4.3.2 MobileFacenet

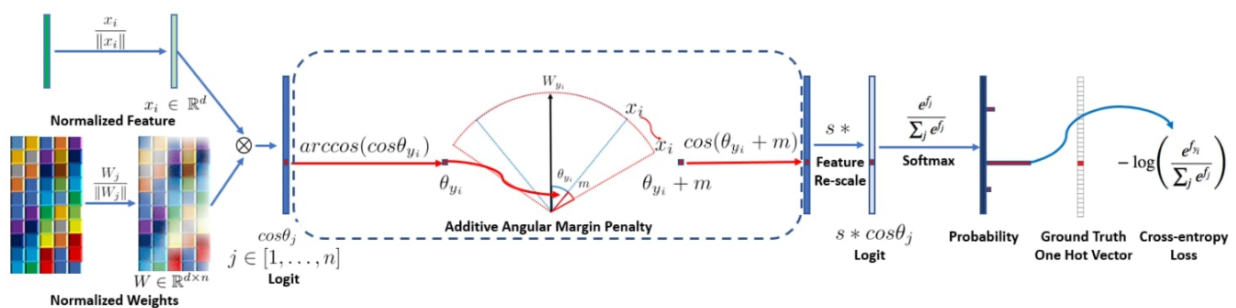
While deep learning based algorithms work exceptionally well, they tend to be very complicated and slow which is a major drawback in constrained environments such as the raspberry pi. Therefore, it may be desirable to sacrifice a bit of accuracy in the favour of a less complicated model that runs faster. The MobileFacenet algorithm is one such algorithm that is specifically designed for high accuracy performance in face recognition in real time. The architecture is similar to that of the standard Facenet model, but the MobileFacenet uses a much smaller feature extractor with only 9 layers and less than 1 million parameters. The MobileFacenet has proven to have performance comparable to that of humans and demonstrates a speed boat of upto 50% over the standard facenet model.

The main improvement in the MobileFacenet comes from replacing the pooling layers. Pooling layers are used in between convolutional layers to reduce the size of the intermediate representation and to effectively condense the intermediate features. The MobileFacenet replaces the global average pooling layer in the feature extractor networks with a depthwise convolution layer which is computationally simpler. Global pooling layers have been observed to be less accurate and deeper networks are needed to compensate for the inefficiency. Depthwise pooling layers achieve the better feature condensation that allows for smaller neural networks that are faster.

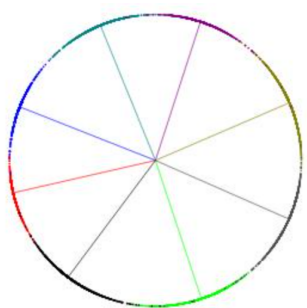
The MobileFacenet seems very promising, therefore, it is included in the potential candidate that are used for experimentation. However, despite the innovations, the MobileFacenet is primarily focused on detection speed and the compactness of the model which means some sacrifices in the performance of the algorithm are to be expected. This, however, may not be a problem if the MobileFacenet achieves satisfactory performance on the raspberry pi.

4.3.3 ArcFace

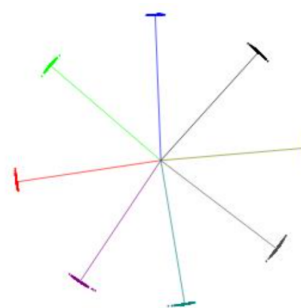
ArcFace is one of the most recent algorithms that was proposed in 2018 in the paper “ArcFace: Additive Angular Margin Loss for Deep Face Recognition”. Like facenet, ArcFace builds on existing convolutional networks and introduces a novel loss function and training mechanism. The Arcface network uses a standard VGG object recognition model as the feature extractor that is finetuned using the ArcFace loss function.



The ArcFace loss function normalized the weights such that the intermediate feature range from -1 to 1. The logit for each category is represented by the angle between the actual truth and the predicted feature inside a sphere. The goal is to ensure that not only losses for non matching faces are high but also far away from the decision boundary. This yields a model that is not extremely sensitive to the threshold and there is less overlap between the classes which leads to cleaner decision boundaries and more robust prediction. The loss function uses two additional hyper parameters, namely the additional angular margin (m) and the scaling ratio for the logit (s) that can be used to adjust the tradeoff between the accuracy and the distance between two classes. A geometric illustration of the principle can be observed below. As you can see, the ArcFace leads to a better separation with less overlap and higher distances between the classes.



(a) Softmax



(b) ArcFace

While the standard softmax function is able to roughly separate the various classes, the distances are uniform which means it has a hard time distinguishing similar looking faces that are close to the decision boundary. The ArcFace overcomes this limitation and is able to distinguish similar looking faces more effectively.

4.3.4 CosFace

CosFace is another example of a loss function that is built specially for face recognition. Cosface was first proposed by researchers at tencent in 2018. Like the networks previously discussed, CosFace also uses pre-existing object recognition

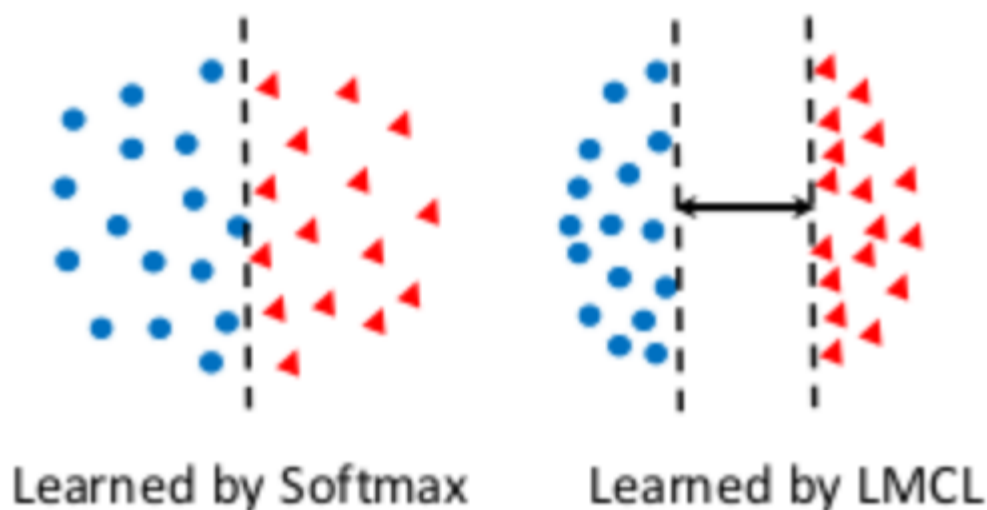
models as feature extractors with the Resnet or the Senet models being the recommended choices.

The core concept in the ArcFace model is the Large Margin Cosine Loss (LMCL) which restructures traditionally used softmax loss by normalizing the intermediate features and the model weights in order to remove the radial variation. The objective is the same as the ArcFace loss and the CosFace model aims to maximize the distances between classes and the decision margin to get cleaner decision boundaries.

First, the weight vectors are standardized using L2 normalization for more effective learning. This means that the prediction only relies on the cosine angle with the norm of the feature vector playing no role. This enables the model to identify features that can be separated reliably in angular space which emphasizes the maximization of accuracy. In addition to the normalized softmax loss, the concept of a cosine margin (m) is introduced to maximize the decision boundary. The cosine margin is a hyperparameter that punishes small classification margins during training which maximizes the distances between classes. The formal CosFace loss function is as follows:

$$L_{lmc} = \frac{1}{N} \sum_i -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j \neq y_i} e^{s \cos(\theta_{j,i})}}$$

An illustration of the CosFace can be observed below. Similar to the ArcFace loss, the CosFace loss enables the face recognition model to better separate the classes which overcomes limitations of the traditional softmax activation function and allows the model to better distinguish the similar looking faces and make more robust predictions.



4.3.5 VGG-Face2

The VGG-Face2 model is an improvement over the older VGG-Face model that was first proposed in 2015 by researchers at University of Oxford. The VGG-Face2 model is architecturally similar to the Facenet models and uses the standard triplet loss function for the model training. As mentioned before in the Facenet subsection, the standard triplet loss uses an anchor, a positive image and a negative image and seeks to minimize the distance between the anchor and the positive image while simultaneously maximizing the distance between the anchor and the negative images.

The main continuation of the VGG-Face model family is arguably the collection and curation of the large dataset that allows for large scale model training. The resnet, VGG or the SqueezeNet is recommended as the best feature extractor networks for the face recognition models. The authors train the three models from scratch and do an extensive evaluation to compare them. The SqueezeNet is observed to have performed better on both face verification and face identification tasks. On the other hand, the resnet models seem to be faster which may be desirable considering the hardware constraints that are posed by the raspberry pi. The authors have open sourced the carefully curated VGG Face dataset which has since been used extensively in several state of the art face recognition.

Section 5 - Face Verification

In the first phase of the testing of the face recognition, the face verification performances of five convolutional neural networks, namely, Facenet, ArcFace, VGG-Face2, MobileFacenet and CosFace recognition algorithms were analyzed. The entire testing infrastructure and the structures of the algorithms were implemented in the python programming language using various deep learning libraries such as keras, pytorch and tensorflow. Pretrained model weights were found and used to run the verification workflow.

Some of the algorithms like Facenet are closed source with no code or models provided by the original authors so third party implementations were used. Dozens of third party models were tested and the ones with the best performance were selected for the comparison. Since this phase of the experiments focuses exclusively on the face verification, the same face detectors, landmarks detector and face alignment algorithms were used in the preceding steps. As mentioned in the previous sections, the Hog detector was selected for the face detection and the ERT algorithm was used for landmark predictions.

5.1 Evaluation Methodology

The tests were run separately on the LFW and the CACD datasets. A large number of tests were done and the data was collected. In each test, one random image was selected from the dataset. A different image of the same person or an image of a different person was chosen with a 50-50 percent probability ensuring there are roughly equal number of positive and negative examples in the test set. The images were loaded and sorted using data ingestors written in python and all images were resized to 512x512 dimensions. Then, the hog face detector and the ERT landmark detector was run and the resultant face landmarks were used to align the extracted bounding box. Subsequently, the cropped face was rescaled into multiple sizes as the various face recognition algorithms are configured to work with different sizes. Finally, the face embeddings were generated using each of the

five algorithms and the euclidean distance was calculated. Along with the distance, the ground truth, prediction of each algorithm, age, gender, race of each person and the verification time was recorded in a CSV file and analyzed in a jupyter notebook. In total, 3000 negative image pairs and 3000 positive image pairs were processed for each dataset which means a grand total of 60,000 verifications were carried out.

5.2 Threshold Selection

The last step of the verification is converting the numeric euclidean distance between the two face embeddings into a boolean prediction. This is done so by setting a maximum distance threshold. Any pair of images with an embedding distance lower than the threshold are predicted to be the same person and a pair with an embedding distance higher is deemed to be of different people. The threshold is selected using a series of trial and error experimentations. In this case, the initial thresholds were selected by manually analyzing the distances of similar and dissimilar pairs. Then an incrementing sequence of thresholds based on the initial threshold were tested and the accuracy, precision and recall metrics were calculated. Finally, the threshold that yields the best performance metrics were selected. The final thresholds are as follows:

Algorithm	Threshold
Facenet	1.12
MobileFacenet	1.25
VGG-Face2	181.3
ArcFace	0.648
CosFace	1.24

These thresholds are used to generate the final prediction and perform the evaluation of comparison of the algorithms in different situations.

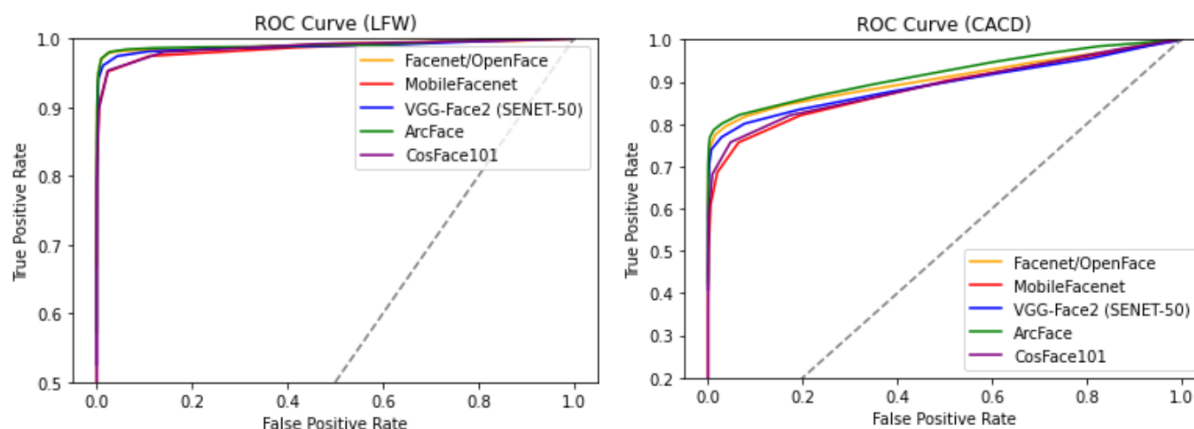
5.3 Initial Results

In addition to the accuracy, precision and recall, the absolute number of true / false positives and negatives along with the false positive rate are also demonstrated. The initial results of algorithms are as follows.

dataset	algorithm	accuracy	precision	recall	FP_rate	n_rows	true_positives	false_positives	true_negatives	false_negatives	time (s)
LFW	ArcFace	0.977	0.974	0.981	0.027	6214	3105	82	2966	61	1.594
	CosFace101	0.965	0.977	0.954	0.023	6213	3018	71	2977	147	0.973
	Facenet/OpenFace	0.978	0.979	0.979	0.022	6214	3098	66	2982	68	0.599
	MobileFacenet	0.964	0.978	0.952	0.023	6214	3014	69	2979	152	0.510
	VGG-Face2 (SENET-50)	0.974	0.987	0.961	0.013	6214	3042	39	3009	124	0.667
CACD	ArcFace	0.887	0.985	0.787	0.012	5906	2321	36	2919	630	1.593
	CosFace101	0.855	0.941	0.757	0.047	5905	2234	140	2814	717	0.971
	Facenet/OpenFace	0.880	0.981	0.775	0.015	5908	2289	45	2910	664	0.597
	MobileFacenet	0.846	0.922	0.757	0.064	5908	2234	190	2765	719	0.492
	VGG-Face2 (SENET-50)	0.870	0.964	0.769	0.029	5907	2271	85	2870	681	0.668

All of the algorithms show promising overall performance on both datasets inline with our expectations. The performance on the easier LFW dataset is phenomenal with the FaceNet and the ArcFace algorithms exceeding human performance (97.2%) at 97.8% and 97.7% respectively. The other algorithms perform in the same ballpark with Mobilefacenet performing the worst by a small margin at 96.4%. The precision and recall metrics tell the same story with metric values in high nineties across the board. On the other hand, the CACD dataset is arguably a lot harder with the accuracies, precision and recalls in the high to mid eighties. ArcFace and Facenet outperform the other algorithms again by a more pronounced margin while the MobileFaceNet has the worst (but still satisfactory and comparable performance). The CACD dataset is hard, even for humans, with the human performance in the sub 90s. This is due to the large variation in the ages in the different images of the same person. This is further reflected by the large number of false negatives which proves the algorithms struggle with positive image pairs due to the drastically different appearances in the images that are often decades apart.

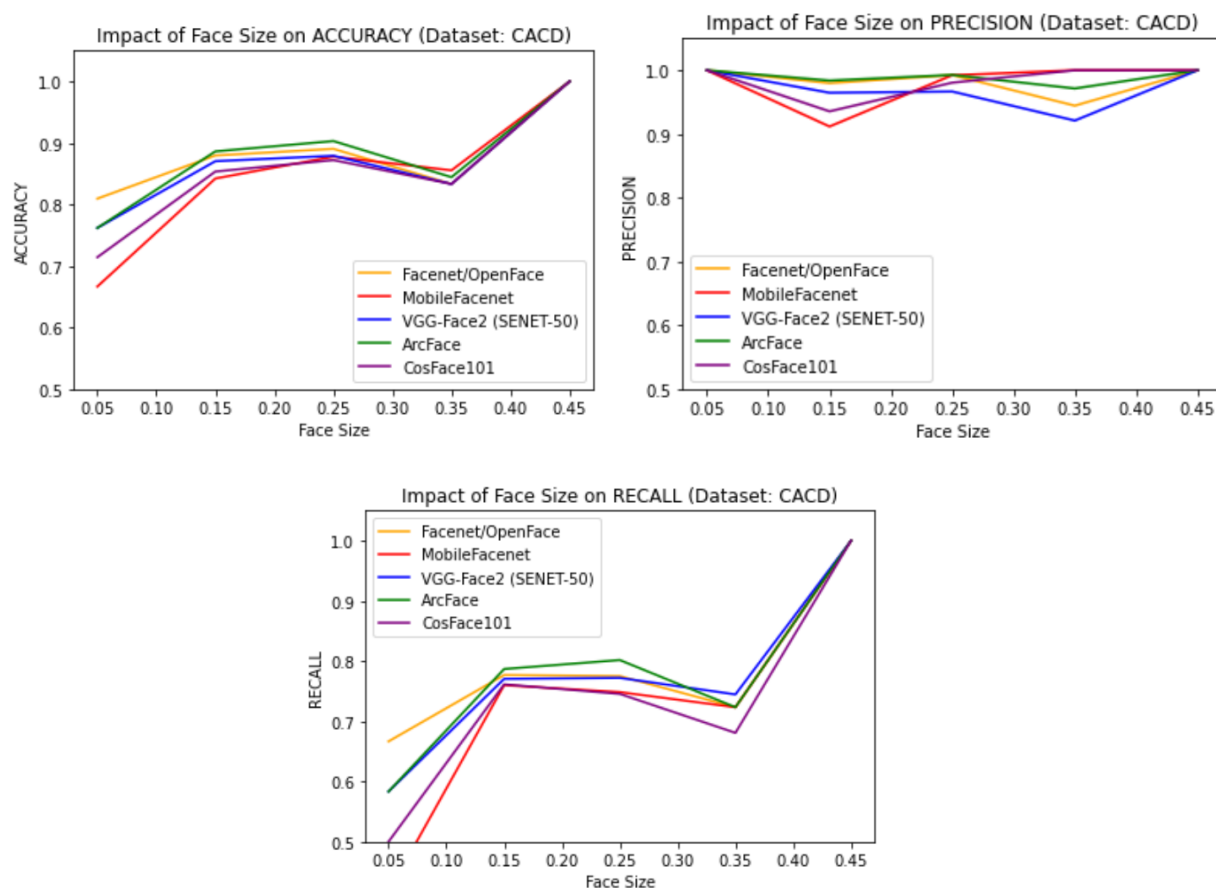
The relative comparisons between the precision and recall as well as between the number of false positives and the number of false negatives illustrate the preferences of each model. At the selected threshold, all of the algorithms tend to act conservatively by favouring false negatives while reducing false positives which is arguably more desirable than the alternative. This behaviour, however, can be tweaked by adjusting the threshold to tweak the “generosity” of the predictions. The tradeoff between the false positives and the false negatives can be illustrated using the receiver operating characteristic (ROC) curve. To calculate this, various thresholds from 0 to 3X the ideal threshold were tested in small increments and the the number of false positives and false negatives were tracked and graphed. The tradeoff is illustrated in the ROC graphs below.



As expected, the facenet and the ArcFace consistently outperform the other algorithms at most of the threshold levels. While the ArcFace has the best performance, it is also the slowest algorithm taking over 1.5 seconds for embedding calculations on a raspberry pi. The Facenet on the other hand is tied with the ArcNet for the best performance and is the second fastest algorithm taking just 0.6 seconds to run. Based on the preliminary results, the FaceNet algorithm is the preferred choice for face verification. Further testing is done and discussed in the subsequent sections in order to evaluate the algorithms in different circumstances for a more thorough analysis.

5.4 Impact of Face Size

The first and most obvious avenue for further analysis of the algorithm is the size of the faces. Simpler frontal faces are easy to deal with and even simpler, non deep learning algorithms perform reasonably well on them. The ideal face recognition algorithm should also perform well on smaller, more difficult faces. To test the impact on the faces sizes, the results were segregated into ten portions based on the proportion of the area of face to the area of the image. For this experiment, only the CACD dataset was considered as there is greater variation in face sizes and the faces are generally harder to deal with. The overall trends in the accuracy, precision and recall are as follows:



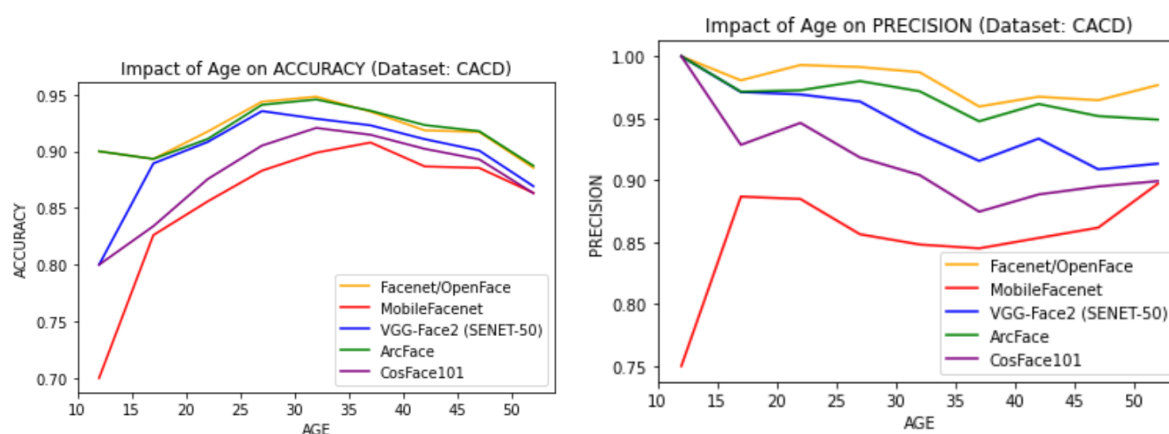
As suspected, the accuracy and the recall (that, in turn, depends on the number of false negatives) rises as the face size increases. The precision stays constant suggesting that the issue with small faces is the number of false negatives, not the

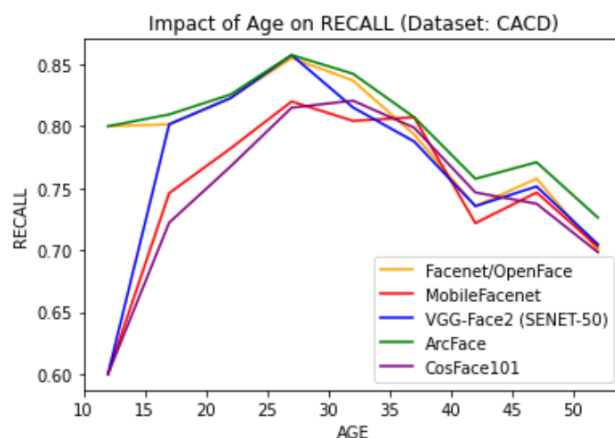
false positives. This is good news as the cost of a false positive is higher than a false negative and prediction conservativeness is a desirable property. At large faces, all algorithms perform near perfectly while the facenet better maintains its' performance as the face size drops and significantly outperforms the other algorithms. For example, for images where faces cover less than 5% of the image, the FaceNet has an 81% accuracy and a 0.68 recall compared to an accuracy of 67% and a 0.38 recall for MobileFaceNet, the worst performing candidate. FaceNet algorithm comes out on top as the preferred choice in this section as well.

5.5 Impact of Age and Difference in Ages

The impact of age on the quality of the predictions is very noticeable while manually checking the results of the verification. This warrants further investigation into the trends in the performance metrics across different ages as well as the relative age difference between the two people being compared.

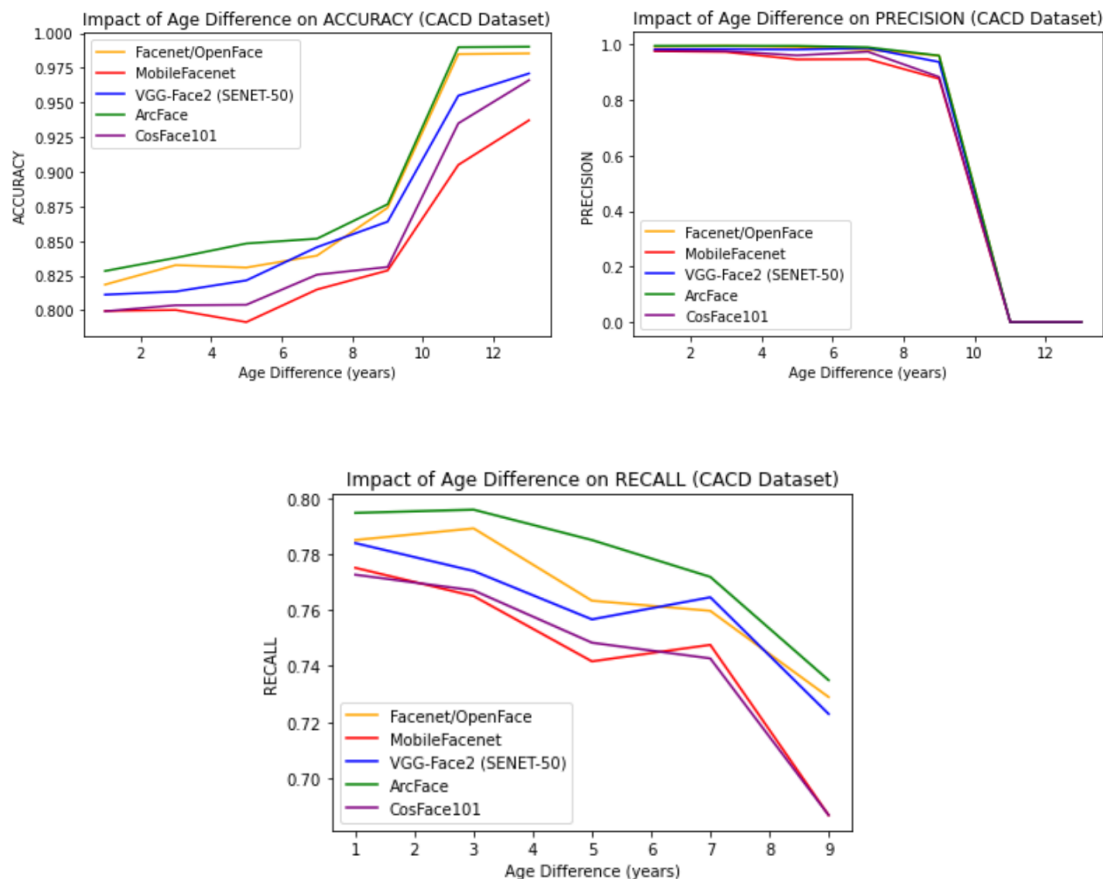
To test the impact of age on the prediction quality, the results were separated into ten separate portions each representing an increasing age bracket. The CACD dataset was used for this test as there is more variance in the ages of the people in the images and there are more images per person making it easier to test the performance on both similar and dissimilar pairs. For each of the age brackets, the performance metrics were observed and analyzed. The general trends in the accuracy, precision and recall are as follows:





According to the results, all of the algorithms perform the best on people in their twenties and thirties while the performance suffers for both younger and older people. An explanation for this phenomena is the fact that the people in the 20 - 40 age bracket comprise the largest proportion of almost all of the face recognition datasets. The median age in the CACD dataset, for example, is 35 years which is consistent with our theory. Fortunately, the algorithms maintain their precision as the age changes which is good news because it means we do not have to worry about false verifications in the older and younger population. Similar to our earlier analysis, the ArcFace and the FaceNet algorithms perform the best across all age brackets. While the other algorithms perform comparably in the 20-40 age bracket, their performance is significantly worse for younger and older people. For example, the FacNet has an accuracy of 90% on ten years old children while the MobileNet, VGG-Face2 and the CosFace algorithms have an accuracy of 80%, 80% and 70% respectively for the same age bracket.

In addition to absolute ages, the difference in the ages of the people being compared also seems to have a noticeable impact on the verification performance. To test this, the age differences for each verification pair were calculated and the dataset was divided into different segments on the basis of the age difference. This experiment was also run on the CACD dataset due to greater variation in ages, more images per person and more difficult images that represent the more complex scenarios. The tests were done on each segment and the results were analyzed



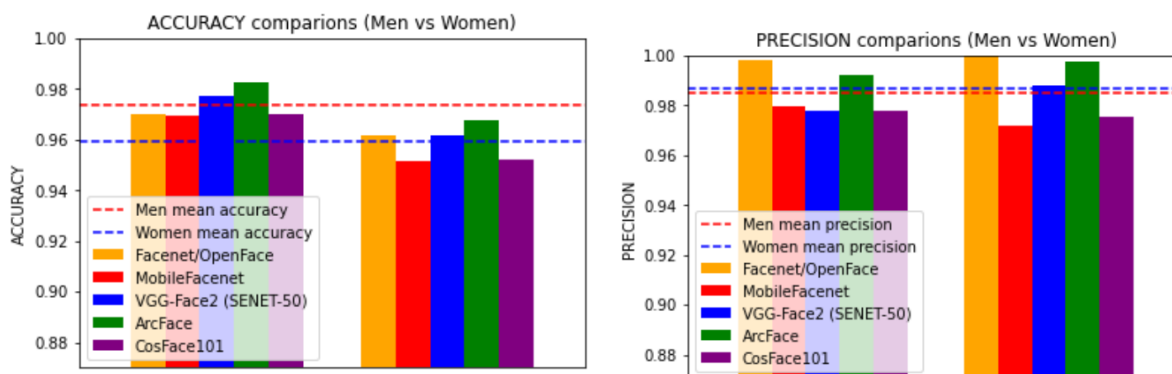
The accuracy seems to show an upwards trend as the age difference rises. This is due to the fact that it is easier to tell people of different ages apart leading to a negligible number of false positives. In addition, there are not a pair of images of the same person a large number of years apart which means there are less false positives (due to less comparison overall) leading to the increase in accuracy. Surprisingly the precision and the recall have the opposite, downwards trend. Further investigation reveals that this is due to the small number of similar pairs at high ages differences leading to a smaller number of true positives (the numeration in both precision and recall). The lack of any true positives (and similarly, no false negatives) at an age difference of above 12 years causes the precision to fall to zero and recall to become undefined. Therefore, the increase in accuracy is slightly misleading and it is fair to conclude that all of the algorithms work better when the age difference is lower.

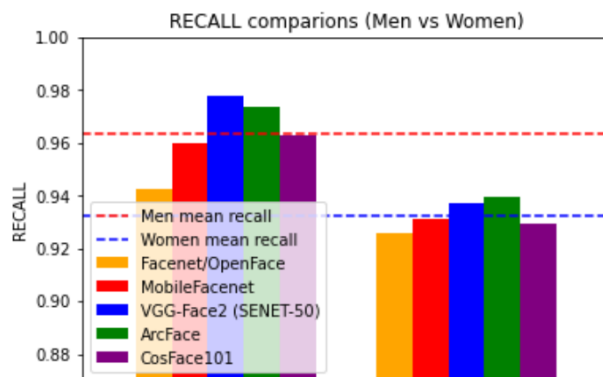
While the individual trends are similar for all of the algorithms, the FaceNet and the ArcFace algorithms outperform the rest while the MobileFaceNet and the CosFace algorithms demonstrate the worst performance at most of the brackets.

5.6 Performance Evaluation for Different Genders

One of the biggest concerns about face recognition is the inherent bias of the algorithms. The deep learning networks learn autonomously but in the process of doing so, they also learn human biases and prejudices that are unconsciously reflected in the training datasets. One of the biases is gender. Images of men are more prevalent comprise a larger proportion of the images in most of the datasets. It is important that a face recognition algorithm is fair and works equally well on both genders.

In order to test this theory, an additional dataset was found that contains the gender information for the LFW dataset. Unfortunately a similar additional datasets that corresponds to the CACD dataset was not found so the experiment was run on only the LFW dataset. A preliminary inspection confirms our suspicion about the gender bias as 73% of the images in the LFW dataset belong to men while only 27% belong to women. The dataset was divided into two segments and the experiments were run separately on both. In addition to manual analysis, a statistical test was run to compare the accuracy in both populations to confirm the existence of a gender bias in the face recognition algorithms. The initial results are illustrated in the graphs below:





All of the algorithms seem to show a gender bias as they demonstrate the worst accuracy and recall on images of women. The precision is similar for both men and women which indicates that the algorithms produce more false negatives on images with women. The overall median accuracy for men in the dataset is 97.4% compared to just 96.3% for women. Likewise the median precision is 97.8% and 98.2% for men and women respectively. The recall has the largest difference as the recall for men is 97.2% compared to just 94.5% for women. In fact, the worst algorithm has a better performance on men than the performance of the best algorithm on women indicating a significant gender bias.

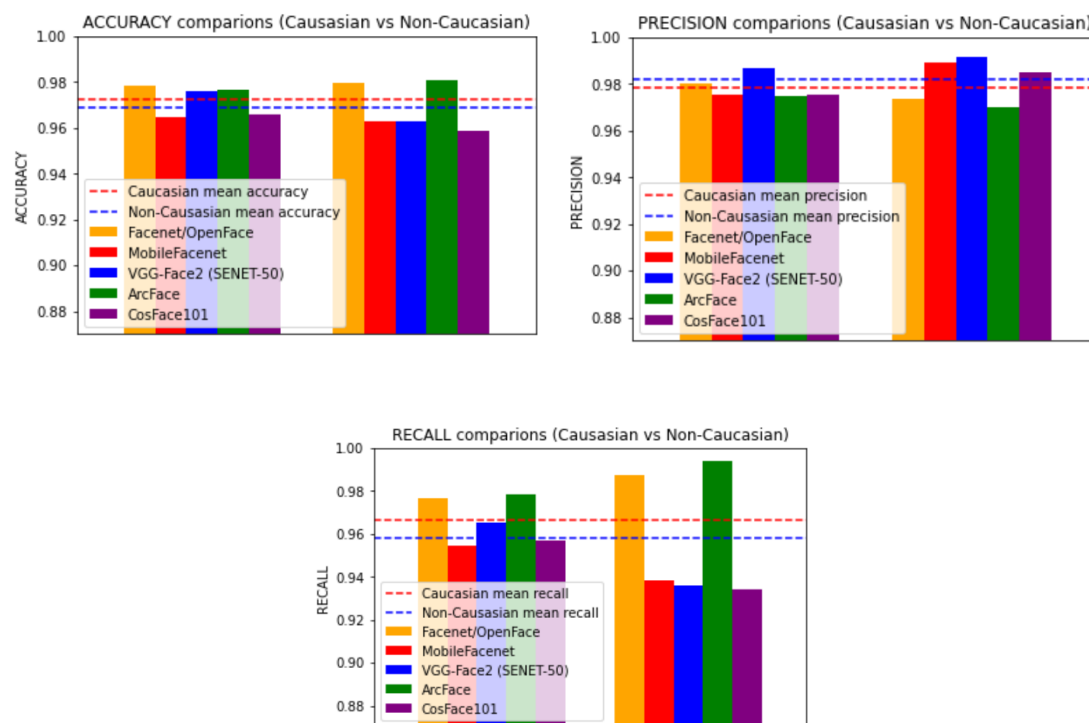
This is further confirmed by a statistical t-test that compares the accuracies of the two populations. The null hypothesis states that there is no difference between the performance of the algorithms on men and women while the alternative hypothesis states that the difference in performance is statistically significant. The test yields a t-statistic of 5.166 and a p-value of 0.00002. This means we can reject the null hypothesis with a very high confidence (>99.9%) and safely conclude that there is, in fact, a difference in performance that warrants further consideration in future research in face recognition.

5.7 Performance Evaluation for Different Ethnicities

Similar to different performances in gender, the difference in performances in different ethnicities is a cause of concern. Ethical critics of face recognition allege that modern face recognition algorithms reflect human biases by performing better

on people of caasian heritage compared to non-causacuian people. This is understandable as most of the face recognition datasets are collected in first world countries that happen to be predominantly caucasian. A larger portion of the datasets contains images of causacian people which, in turn, translates to a worse performance for non- caucasian people. This is undesirable and an ideal face recognition algorithm should not display this behavior.

A supplementary dataset was found that contains a binary variable corresponding to ethnicity of subject of each image in the LFW dataset. Unfortunately a similar additional datasets that corresponds to the CACD dataset was not found so the experiment was run on only the LFW dataset. A preliminary inspection confirms our suspicion about the ethnic bias as 84% of the images in the LFW dataset belong to people with caucasian heritage while only 16% belong to non-caucasuian people. The dataset was divided into two segments and the experiments were run separately on both. In addition to manual analysis, a statistical test was run to compare the accuracy in both populations to confirm the existence of an ethnic bias in the face recognition algorithms. The results of the experiments are illustrated in the graphs below:



Despite the understandable concerns, no significant difference is observed in the performances of the face recognition algorithms. Some of the algorithms perform marginally better on caucasian people but the difference is miniscule at a fraction of a percent. Some of the algorithms such as the FaceNet and the ArcNet even (slightly) perform better on non-caucasian people. This holds true for the accuracy, precision and recall metrics and the performance differences seem so insignificant that they can simply be attributed to statistical noise.

The initial observation is further confirmed by a statistical t-test that compares the accuracies of the two populations. The null hypothesis states that there is no difference in the performance of algorithms on caucasian and non-caucasian subjects while the alternative hypothesis states otherwise. The test yields a t-statistic of 1.27 and a p-value of 0.2. This means we fail to reject the null hypothesis at any reasonable confidence level and can safely conclude that there is no significant difference in the performances on caucasian and non-caucasian subject. Running the t-tests collectively and individually on each algorithm yields the same result and the null hypothesis holds true in all cases.

5.8 99.9% True Positive Rate

Until now, the false positives and the false negatives are treated similarly and the thresholds are selected to maximize the overall accuracy. This may not be the ideal path to take as the consequences of a false positive result are a lot higher as it means granting unauthorized access. Conversely, a false negatives means incorrectly denying access which is a less severe offense. This means that the thresholds should be selected to minimize the false positive rate even if it translates to a higher false negative rate and a lower overall accuracy. The general tradeoff is observed by the ROC curve illustrated in the preceding sections. The ideal threshold can be selected by setting a minimum precision (percentage of positive predictions that are actually true).

A symbolic objective for face verification algorithms is a precision of 99.9%. This translated into a false positive rate of less than 0.1% which means, on average, less

than 1 out of every 1000 positive predictions are wrong. This is also applicable to the real world application at hand roughly 1 unauthorized access per thousand verifications seems acceptable. In order to identify the correct thresholds, an experiment was designed which involves starting from a low threshold and testing incrementally increasing threshold levels until the 99.9% precision is observed. The minimum thresholds for a 99.9% precision are reported in the table below.

Algorithm	99.9% Precision Threshold
Facenet	0.941
MobileFacenet	1.075
VGG-Face2	152.2
ArcFace	0.557
CosFace	1.091

The new thresholds are lower than the previous thresholds indicating that the algorithms are more conservative and more stringent with positive predictions. The new thresholds warrant another investigation in the predictions of the algorithms on each dataset. The metrics are calculated again using the new thresholds and are reported in the following table.

dataset	algorithm	accuracy	precision	recall	FP_rate	n_rows	true_positives	false_positives	true_negatives	false_negatives	time (s)
LFW	ArcFace	0.956	0.999	0.914	0.001	6214	2893	3	3045	273	1.594
	CosFace101	0.860	1.000	0.726	0.000	6213	2297	0	3048	868	0.973
	Facenet/OpenFace	0.946	0.999	0.895	0.001	6214	2832	2	3046	334	0.599
	MobileFacenet	0.859	1.000	0.722	0.000	6214	2287	0	3048	879	0.510
	VGG-Face2 (SENET-50)	0.913	1.000	0.829	0.000	6214	2625	1	3047	541	0.667
CACD	ArcFace	0.857	1.000	0.713	0.000	5906	2105	0	2955	846	1.593
	CosFace101	0.772	0.999	0.544	0.000	5905	1605	1	2953	1346	0.971
	Facenet/OpenFace	0.832	1.000	0.664	0.000	5908	1960	0	2955	993	0.597
	MobileFacenet	0.757	0.998	0.514	0.001	5908	1519	3	2952	1434	0.492
	VGG-Face2 (SENET-50)	0.822	0.999	0.644	0.000	5907	1900	1	2954	1052	0.668

In order to achieve the coveted 99.9% precision, recall and overall accuracy has to be sacrificed and the false negatives increase significantly for most of the algorithms. The differences between the error metrics of the various algorithms are much more pronounced in the new results.

In line with our previous results, the ArcFace and the FaceNet algorithms perform the best and only lose 2% and 3% of the accuracy on the LFW dataset respectively compared to the MobileFaceNet and the CoseFace that lose 10% of their accuracies. Similarly, the total number of false positives rises more drastically for the worse performers. For example, the MobileFaceNet predicts 879 false negatives on the LFW dataset (compared to 124 false negatives before) in contrast with the 273 false negatives (compared to 61) for ArcFace. This means that not only are the FaceNet and ArcFace algorithms better predictors, their predictions are also further away from threshold (in either direction) which leads to more robust and stable predictions.

5.9 Conclusion

The wide array of tests and experiments provide very conclusive results. The FaceNet and the ArcFace algorithms outperform the other algorithms in most of the tests and perform better in almost all of the segments in the individual tests. Conversely, the CoseFace and the MobileFaceNet performs the worst while VGG-Face2 ranks in between. Therefore, the final selection should be made between ArcFace and FaceNet. While the ArcFace algorithm nominally outperforms the FaceNet algorithm, it is almost 3 times slower as it takes roughly 1.5 seconds for a single pass on a raspberry pi compared to just 0.6 seconds for the FaceNet. Since the speed of recognition is important, the FaceNet algorithm presents itself as the ideal choice for our application. In fact, the FaceNet has statistically similar performance to that of the best algorithms in terms of accuracy (ArcFace) as well as speed (MobileFaceNet) as the MobileFaceNet is only marginally faster and offers significantly worse performance.

Finally, the 99.9% precision threshold is selected for the final predictions in order to give precedence to the reduction of false positives at the expense of false negatives. The thresholding mechanism is further refined in the face identification evaluation with the introduction of dynamic thresholding.

Section 6 - Face Identification

In the second phase of the testing of the face recognition, the face identification performances of five convolutional neural networks, namely, Facenet, ArcFace, VGG-Face2, MobileFacenet and CosFace recognition algorithms were analyzed. Face identification is the generalized form of face verification. It refers to the problem of recognizing the face of a human from a predefined collection of human faces. The outputs of face recognition are two folds. The first goal is to determine if the face in question is recognized (that is, matches one of the faces in the collection) or is unknown. The second goal is the determination of the person, if any, the face belongs to. Contrary to face verification, face recognition is a one-to-many problem. Unsurprisingly, the large number of pairwise comparisons and the two fold objectives mean that face recognition is significantly more complex compared to standard pairwise verification. The complexity rises as the number of images in the test dataset increases. A larger number of images in the test dataset increases the probability of similar faces which increases the false positives. Face identification is more useful in the real world and is more relevant to our real world application which means the identification results should be prioritized over the verification results.

The entire testing infrastructure and the structures of the algorithms were implemented in the python programming language using various deep learning libraries such as keras, pytorch and tensorflow. Pretrained model weights were found and used to run the verification workflow. Some of the algorithms like Facenet are closed source with no code or models provided by the original authors so third party implementations were used. Dozens of third party models were tested and the ones with the best performance were selected for the comparison. Since this phase of the experiments focuses exclusively on the face verification, the same face detectors, landmarks detector and face alignment algorithms were used in the preceding steps. As mentioned in the previous sections, the Hog detector was selected for the face detection and the ERT algorithm was used for landmark predictions.

6.1 Evaluation Methodology

The tests were run separately on the LFW and the CACD datasets. A large number of tests were done on various dataset sizes (number of images in the test dataset) and the data was collected. In each test, a random section of the dataset was selected. The dataset sizes vary from just 10 images to 30,000 images. One random image was selected and removed from the dataset. In half of the experiments, all images of the selected person were removed from the dataset simulating the unknown person scenario. For the other half of the experiments the other images of the selected were left in the dataset simulating the known person scenario.

The face recognition pipeline was run for the selected images along with all of the images in the testing dataset. The embeddings of the original image were compared with all embeddings of the testing dataset and the image with the minimum distance to the selected image was selected. Finally, the minimum distance is compared to thresholds in order to make the final prediction.

Variables such as the minimum distance, time taken, dataset size, number of images of the selected person and the ground truth were recorded in a CSV file and analyzed in a jupyter notebook. For each dataset and algorithm, 20,000 experiments were done at various dataset sizes. In total, about 225,000 experiments were carried out,

6.2 Threshold Selection

Just like the last step of face verification, the euclidean distances have to be converted into boolean predictions by utilizing a maximum distance threshold. Any pair of images with an embedding distance lower than the threshold are predicted to be the same person and a pair with an embedding distance higher is deemed to be of different people. The initial thresholds were selected by manually analyzing the distances of similar and dissimilar pairs. Then an incrementing sequence of thresholds based on the initial threshold were tested and the accuracy, precision

and recall metrics were calculated. Finally, the threshold that yields the best performance metrics were selected.

Face recognition typically yields a higher number of false positives and a lower accuracy due to the one to many nature of the comparison. Therefore, it is reasonable to assume the final thresholds are going to be lower than the ones observed in face verification. This means that the algorithms are going to behave more conservatively. The ideal thresholds observed are as follows:

Algorithm	Threshold
Facenet	0.82
MobileFacenet	1.0
VGG-Face2	136.9
ArcFace	0.49
CosFace	1.0168

These thresholds are used to generate the final prediction and perform the evaluation of comparison of the algorithms in different situations. It is reasonable to assume that the larger the dataset grows, the ideal threshold falls as the ROC curve shifts inwards. This can be exploited by dynamic thresholding which is explored further in the subsequent sections.

6.3 Initial Results

In addition to the accuracy, precision and recall, the absolute number of true / false positives and negatives along with the false positive rate are also demonstrated. Furthermore, the number of misclassifications (false positives in examples where the correct person is in the dataset) and incorrect refusals (false negatives in examples where the correct person is in the dataset) are used to get a deeper understanding of the behaviour of the algorithms. The initial results of algorithms are as follows.

dataset	algorithm	accuracy	precision	recall	FP_rate	true_positives	false_positives	true_negatives	false_negatives	misclassifications	misrefusals
LFW	arcface	0.906	0.946	0.863	0.050	10630	607	11580	1690	99.0	1591.0
	cosface	0.804	0.983	0.621	0.011	7647	135	12052	4673	66.0	4607.0
	facenet	0.885	0.950	0.814	0.043	10028	524	11663	2292	107.0	2185.0
	mobilefacenet	0.829	0.975	0.678	0.018	8350	217	11970	3970	96.0	3874.0
	vggface	0.873	0.983	0.761	0.013	9371	158	12029	2949	22.0	2927.0
CACD	arcface	0.854	0.862	0.843	0.134	10242	1641	10593	1914	372.0	1542.0
	cosface	0.805	0.814	0.790	0.180	9605	2202	10032	2551	448.0	2103.0
	facenet	0.834	0.845	0.817	0.149	9935	1818	10416	2221	374.0	1847.0
	mobilefacenet	0.770	0.767	0.774	0.234	9409	2860	9374	2747	546.0	2201.0
	vggface	0.835	0.848	0.815	0.145	9912	1780	10454	2244	339.0	1905.0

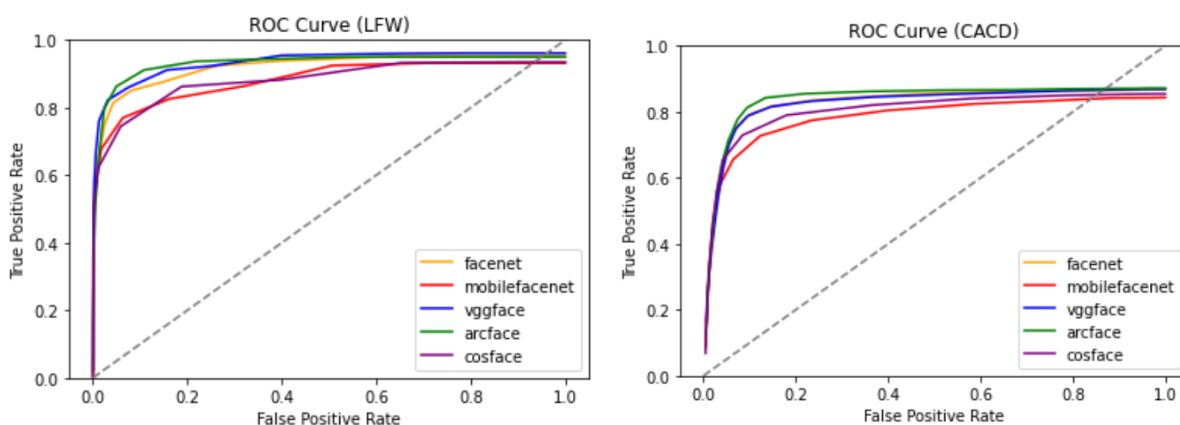
Understandably, all of the metrics are lower than their counterparts in face verification due to added complexity. All of the algorithms show promising overall performance on both datasets inline with our expectations. The same overall trends are observed with the Facenet and the ArcFace algorithms outperforming the rest in all of the metrics. No reasonable human performance could be found in the literature but a cursory review of the errors and the manual observation of the misjudged images with reference to the type of error (misclassification or incorrect refusal) shows comparable performance to humans especially for the ArcFace and the Facenet Algorithms.

The performance on the easier LFW dataset is very good with the accuracies of the Facenet and the ArcFace algorithms at 90.6% and 88.5% respectively. The difference between FaceNet/ArcFace and the worse performing algorithms is more significant compared to the earlier verification results. The CosFace algorithm performs the worst with an accuracy of just 80.2% which means the the comparable verification accuracies do not translate to similar performances on in face identification. The precision and recall metrics tell the same story with metric values in high nineties across the board.

On the other hand, the CACD dataset is arguably a lot harder with the accuracies, precision and recalls in the high to mid eighties. ArcFace and Facenet outperform the other algorithms again by a more pronounced margin while the CosFace has the worst (but still satisfactory and comparable performance). The difference between the CACD and the LFW dataset is less pronounced and the ArcFace and Facenet

algorithms seem to better maintain their performance on more difficult datasets (CACD) indicating higher robustness and stability.

The relative comparisons between the precision and recall as well as between the number of false positives and the number of false negatives illustrate the preferences of each model. At the selected threshold, all of the algorithms tend to act conservatively by favouring false negatives while reducing false positives which is arguably more desirable than the alternative. This behaviour, however, can be tweaked by adjusting the threshold to tweak the “generosity” of the predictions. The tradeoff between the false positives and the false negatives can be illustrated using the receiver operating characteristic (ROC) curve. To calculate this, various thresholds from 0 to 3X the ideal threshold were tested in small increments and the the number of false positives and false negatives were tracked and graphed. The tradeoff is illustrated in the ROC graphs below.



The ROC curves are shifted inwards compared to the ROC curves of face verification due to the higher complexity. A peculiar observation is that the curves do not fully extend fully to the theoretical limits and seem to taper off at a true positive rate of less than 1. This is due to the nature of face identification and the fact that face identification is a pure classification problem with three outcomes rather than just two. In some of the examples, an image of a different person has a lower euclidean distance compared to that of an image of the same person. This means that the person is misidentified and no amount of threshold tweaking will fix the problem which means the algorithm never attains a 100% true positive rate.

This is more pronounced in the CACD dataset in which the age varies more drastically increasing the probability of misidentification.

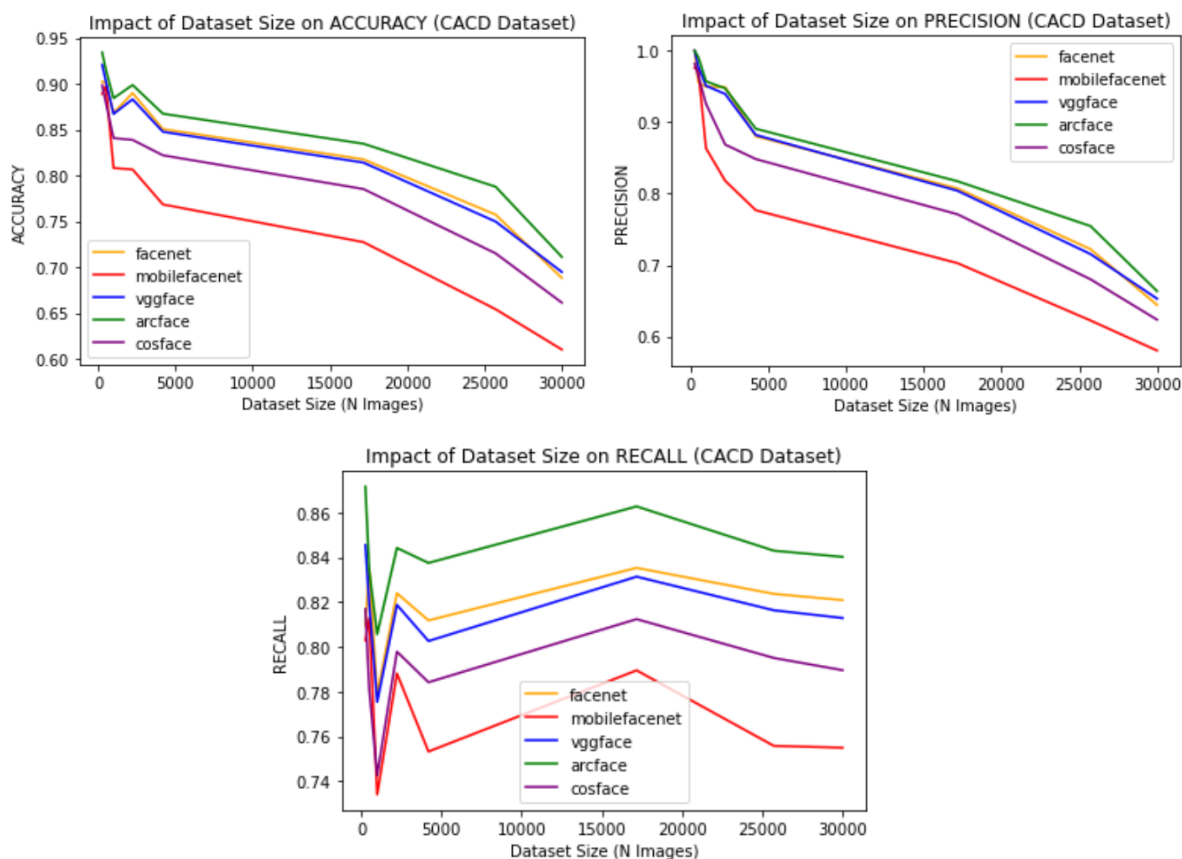
Other than this, the observed curves behave as expected with the Facenet and ArcFace algorithms consistently outperforming the other algorithms at most of the threshold levels. While the ArcFace has the best performance, it is also the slowest algorithm taking over 1.6 seconds for embedding calculations on a raspberry pi. The Facenet on the other hand is tied with the ArcNet for the best performance and is the second fastest algorithm taking just 0.7 seconds to run. Based on the preliminary results, the Facenet algorithm is the preferred choice for face verification. Hence, the initial results corroborate all of the earlier testing results. Some deeper analysis is done in the subsequent sections before the final choice can be made. Results of the experiments made with reference to face sizes, number of faces per image, gender, race, age and age difference mirror the results of similar experiments in face verification, therefore, they are not discussed further. Instead, the impact of the number of images in the test dataset and the impact of the number of images of the selected person are discussed.

6.4 Impact of Dataset Size

Compared to pairwise face verification, face recognition poses the additional problem of selecting the best match out of a dataset with multiple test images. It is reasonable to expect the performance of any face recognition algorithm to drop as the number of images in the test dataset rises. This is simply due to the larger comparisons to be made and the higher probability of a similar, false negative image. In addition to the performance, one can also expect the recognition time to increase linearly with the size of the dataset simply because of the larger number of comparisons.

An experiment was designed in order to test this hypothesis. The general face recognition test was run multiple times using different face recognition algorithms and test dataset sizes and the accuracy was measured for each instance. The results were then grouped by the number of images and performance metrics along with

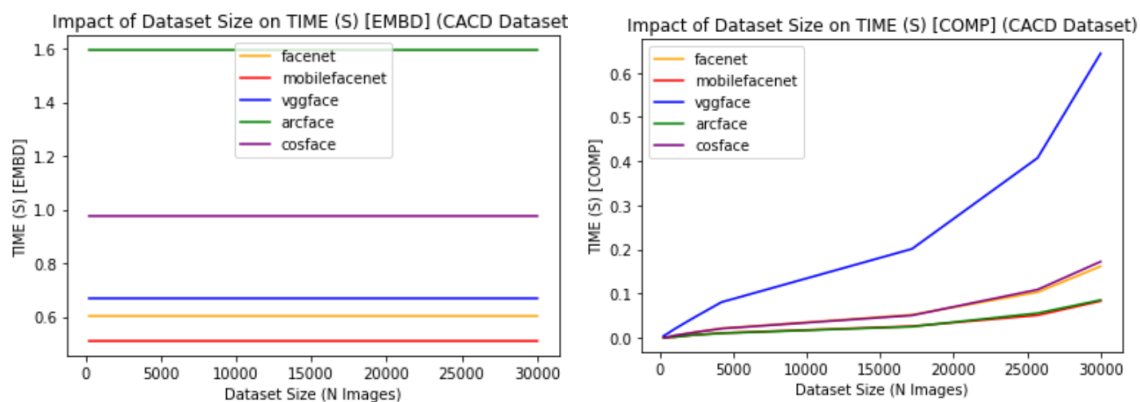
the recognition times were averaged and graphed for demonstration. This experiment was carried out on the CACD dataset due to the greater difficulty that mirrors that of real life scenarios. The results of the these experiments are as follows:



For all algorithms, a downwards trend can be observed in the face accuracy and precision while the recall metric does not change by much. This suggests that the number of false positives rise as the test dataset size increases while the number of false negatives does not change significantly. This is further confirmed by a manual inspection of the mislabelled instances. The algorithms work very well and maintain their performance if the selected person is present in the dataset suggesting the relative comparisons are very robust. The more problematic scenario is when the image of the selected person is not present in the dataset. In that case, the importance of the arbitrary threshold is magnified as it is more likely that a similar looking person is present in a large dataset, hence leading to a larger number of false positives, lower precision and worse accuracy. This can be

somewhat remedied by dynamic thresholding that is discussed in the next section. The ArcFace and the Facenet algorithms win again and perform better than the other algorithm across all dataset sizes and metrics making one of them the obvious choice.

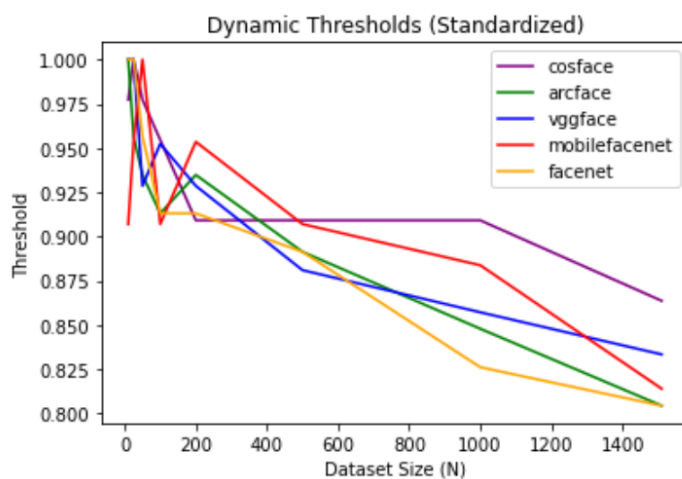
The recognition time is just as important as the performance so the recognition time was analyzed in reference to the test dataset size. The recognition time was broken down into two components. The first component corresponds to the time taken for embedding calculation and the second component corresponds to the time taken for calculations necessary to calculate the distances between embeddings.



Understandably, the time taken for embedding calculation stays the same as a single embedding (that of the selected image) is calculated in real time regardless of the dataset size. Conversely, the comparison component rises linearly as the number of calculations necessary for comparison scale linearly. The VGG-Face scales the worst owing to the largest embedding size (512 dimensional embedding). On the other hand, the ArcFace and the MobileFacenet scale the best due to their 128 dimensional embeddings. That said, at small dataset sizes (less than 30,000 images), the comparison time is much lower than the embedding calculation time making ArcFace the worst algorithm in terms of speed. The scaling advantage only comes into play if the test dataset contains hundreds of thousands of images. The Facenet algorithm seems to be the best compromise as it offers the second best embedding calculation time and the second best scaling in terms of comparison time.

6.5 Dynamic Thresholding

The problem of the large number of false positives in larger datasets stems from the limitations of the static thresholding. Using the same threshold to make the final decision across all dataset sizes tends to be inefficient as it is either too low and results in a large number of false negatives for smaller datasets or it is too high and yields a large number of false positives on larger datasets. An interesting approach to resolve the tradeoff and rectify the situation is dynamic thresholding. In dynamic thresholding, multiple optimal thresholds are used for different dataset sizes instead of a single threshold across the board. In general, a higher threshold is used for smaller datasets to minimize false negatives. The threshold is progressively lowered as the dataset increases in size to offset the increasing risk of false positives. Dynamic thresholding can be configured using a simple decreasing function or the explicit calculation of optimal thresholds and various levels. Both approaches were experimented with and better results were observed with the explicit calculation due to the removal of arbitrary assumptions that come with the mathematical approach. The scaled optimal thresholds for each dataset size can be observed in the following graph.



The explicitly calculated thresholds are stored in a lookup table and can be referenced during the recognition phase before making a final prediction.

6.5 Conclusion

The results in the face recognition experiments corroborate what has already been observed in the preceding section. The Facenet and the Arcface algorithms consistently beat the other algorithms by a significant margin in most of the tests. Not only do they demonstrate higher performance metrics but are also more robust and tend to maintain their performance in difficult scenarios pretty well. While the Arcface algorithm is nominally better than Facenet, it is more than twice as slow. Infact, the only algorithm that is faster than the Facenet is its' less complicated counterpart, the Mobile Facenet which is consistently the worst performer in the majority of the undertaken experiments. Since speed is a major concern due the the hardware constraints posed by using the raspberry pi as the base of the platform, the Facenet algorithm is the obvious choice and is selected for the face verification and face recognition part of the pipeline.

Furthermore, since the overall number of employees is variable, dynamic thresholding was chosen as the classification mechanism in the interest of a more robust system. Explicit threshold calculations were carried out for the Facenet algorithm using a combination of the CACD and the LFW dataset and the optimal thresholds are saved in a CSV file that is referenced by the desktop application.

Section 7 - Web Portal and Desktop App

As the final goal of the dissertation, a complete face recognition system consisting of a centralized web portal, a desktop application for a raspberry pi that connects with the web portal in real time and on click installation scripts for installation and registration of new face recognition nodes (raspberry pis). The goal is to provide an easily configurable, plug and play system that can be setup by non technical people in a few minutes.

At this point, the testing, experimentation and technical implementation of the various algorithms has been completed and one algorithm has been selected for each individual component of the face recognition pipeline. While the implementation and analysis are useful for technical people, considerable effort is required by the layperson to understand all aspects of the technology and put the system in practice. Therefore, the surrounding infrastructure is just as important as the core face recognition technologies. There is a need for an easily configurable, ready made face recognition system that the layperson should be able to use without having to understand the technical aspects.

7.1 Web Portal

A web portal was implemented to manage the workforce and provide a centralized interface to business users. The core web portal is written in python using the django framework. The consumer portal is accessible through any web browser as a standard web application. A second server was written using websockets which is primarily used to manage real time notifications and alerts in the web app. In addition, the websockets server is also used by the desktop application to enable real time communication which allows for instant syncing in case a new employee or image is added to the webportal. The server also exposes an application programming interface (API) which is used to receive authorization requests from the desktop application and returns an approval or rejection based on the user permissions defined in the portal.

For production, the web application is hosted on a digital ocean Ubuntu instance and uses NGINX as the reverse proxy, Gunicorn as the HTTP server, Daphne as the ASGI server for the websockets and Postgresql as the database. Note that all server components are part of the same application and use the same database.

The code can be found in the github repository and a live demo of the web portal can be viewed at the following location:

Url: <https://facerecognition.jazeetech.com/>

Username: test

Password: test123

Some of the features of the web application include:

- Real Time Communication with the desktop application
- Syncing datasets and managing access requests
- Room Management: Adding or removing rooms
- Camera Management: Adding or removing rooms
- Employee Managements: Adding or removing Employees
- Access Permission: Managing employee access to rooms
- Alerts: Realtime notifications for unauthorized verifications
- History: Historical logs of all authorization requested
- Reports: Downloadable attendance reports for employee

In order to better illustrate the feature, a few screenshots of the web portal are included in this section.

The “Rooms” submenu allows us to add or remove new rooms. A room is defined as a single enclosed space in a building. Access permission is granted on the basis of rooms

FACE RECOGNITION

- Rooms
- Cameras
- Employees
- Alerts
- Authorizations
- Reports

Rooms

Name	Activation Code	N Cameras	Date Created	
Room 1	WAYNX	1	Aug. 10, 2021, 9:54 p.m.	✕
Room 2	YDYRO	0	Aug. 10, 2021, 9:58 p.m.	✕
Room 3	SVKOF	0	Aug. 10, 2021, 9:58 p.m.	✕

Activate Windows
Go to Settings to activate Windows.

The “Cameras” submenu is designed to keep track of all of the registered cameras. A camera refers to a physical node (that is, a raspberry pi). A camera is attached to a single room and uses the permissions set for that particular room. On the other hand, one room can have multiple cameras for greater flexibility and to allow for multiple entry points. A camera record is automatically created when a raspberry pi is registered successfully using the one-click installation scripts explained below.

FACE RECOGNITION

- Rooms
- Cameras
- Employees
- Alerts
- Authorizations
- Reports

Cameras

Name	Room	Date Created	Date Synced	Status	
Camera 1	Room 1	Aug. 10, 2021, 9:54 p.m.	Aug. 10, 2021, 9:55 p.m.	DISCONNECTED	✕

Activate Windows
Go to Settings to activate Windows.

Next, the “Employees” submenu allows us to manage employees. The home screen allows for the user to view, add or delete employees while additional interfaces are used to manage the images and permission for an individual employee.

Name	Date Created	Last Seen	# of Pictures	# of Permissions	
Mario Draghi	Aug. 10, 2021, 9:58 p.m.		1	3	
Boris Johnson	Aug. 10, 2021, 9:55 p.m.		1	0	
Angela Merkel	Aug. 10, 2021, 9:54 p.m.		1	1	
Talha Ijaz	Aug. 10, 2021, 9:54 p.m.		3	1	

Talha Ijaz

DELETE DELETE DELETE

Edit Employee ✕

Employee Name
Talha Ijaz

Permissions


Room 1

Room 2

Room 3

SAVE

The “Alerts” and “Authorizations” submenus allow the user to check historical logs of the approvals and rejections given by the system. In case of an unauthorized access attempt, a real time notification is shown on the top left side of the webpage.




FACE RECOGNITION

- 🏠 Rooms
- 📹 Cameras
- 👤 Employees
- ⚠️ Alerts
- 🕒 Authorizations
- 📄 Reports

Alerts

Date	Person	Room	Camera	
Aug. 10, 2021, 9:57 p.m.	Boris Johnson	Room 1	Camera 1	UNAUTHORIZED
Aug. 10, 2021, 9:56 p.m.	Unknown Person	Room 1	Camera 1	UNKNOWN
Aug. 10, 2021, 9:56 p.m.	Unknown Person	Room 1	Camera 1	UNKNOWN
Aug. 10, 2021, 9:56 p.m.	Unknown Person	Room 1	Camera 1	UNKNOWN
Aug. 10, 2021, 9:56 p.m.	Boris Johnson	Room 1	Camera 1	UNAUTHORIZED

Activate Windows
 Go to Settings to activate Windows.



FACE RECOGNITION

- Rooms
- Cameras
- Employees
- Alerts
- Authorizations**
- Reports

Authorizations

Date	Person	Room	Camera	
Aug. 10, 2021, 9:57 p.m.	Angela Merkel	Room 1	Camera 1	AUTHORIZED
Aug. 10, 2021, 9:57 p.m.	Talha Ijaz	Room 1	Camera 1	AUTHORIZED
Aug. 10, 2021, 9:57 p.m.	Talha Ijaz	Room 1	Camera 1	AUTHORIZED
Aug. 10, 2021, 9:56 p.m.	Angela Merkel	Room 1	Camera 1	AUTHORIZED

Activate Windows
Go to Settings to activate Windows.

Finally, the “Reports” submenu allows the user to download authorization records for a particular employee in the form of CSV files.

Reports

Employee

DOWNLOAD REPORT

7.2 Desktop Application

The second component of the system is a local application designed to be run on a raspberry pi to execute the actual face recognition algorithms and work with the web servers to manage accesses and authorizations. A python project was implemented to serve as an always running daemon that performs the actual recognition on a single face recognition node and communicates to grant or deny accesses. Some of the features of the Desktop Application are as follows:

- Real time communication with the server;
- Syncing employees images;
- Performing continuous face recognition;
- Face recognition on trigger from the raspberry pi GPIO pins;
- Requesting access from server and manual;
- Triggering GPIO pins to simulate actuator control.

The desktop application has no graphic user interface and works autonomously. After the initial installation, no further human intervention is required.

7.3 One Click Install

To further facilitate setup, the “batteries included” concept is used. First, a Raspbian OS ISO image is created with all of the necessary software including python installation, python packages, the local application and a supervisor service to automatically start the application on boot up and keep it running forever in the background. Thus, converting a raspberry pi into a camera node is as simple as a simple OS installation which includes everything.

After OS installation, one simply has to run the “install.py” file in the root directory and enter the room activation code (from the web portal) when prompted. The script takes care of the rest of the installation and automatically adds the camera node and syncs the images stored on the cloud. At this point, the camera node is activated and the raspberry pi can be left alone to work autonomously.

Section 7 - Conclusion

The dissertation is comprehensive and all of the original goals have been accomplished successfully. A single reading of the report should be enough to provide a complete introduction to face recognition. The pipeline is explained and each individual component is introduced and discussed in detail. For each component, the research landscape is investigated including the historical and modern approaches. Finally, a set of promising algorithms is selected to be tested in the experimental section of the dissertation. A literature review has been written to explain the selected algorithms for each individual part of the face recognition pipeline.

The experimental portion of the dissertation deals with the experiments that were designed to evaluate the different algorithms from various perspectives and to choose the best suited algorithm for each phase of the face recognition pipeline. The algorithms were implemented successfully, tested and experimented thoroughly to analyze their performance under various circumstances in general and in the context of our specific employee access control application. The strengths and weaknesses of the algorithms were discussed and a single algorithm has been selected for each phase. The final selection for unconstrained face recognition on a raspberry pi as as follow:

Face Detection: Histogram of Oriented Gradients (HOG) Detector

Landmark Detection: Ensemble of Regression Trees (ERT) Detector

Face Alignment: 2D face alignment with image rotation and centering

Face Recognition: FaceNet Convolutional Network with dynamic thresholding

Lastly, an end-to-end plug and play face recognition platform was successfully developed which includes a centralized web portal, a desktop application, OS disk images and installation scripts with all the batteries included.

As a final thought, I think the project is very detailed and the dissertation does a good job explaining the undertaken project. The developed infrastructure makes it incredibly easy to implement a face recognition employee authorization system in

office buildings and can be configured easily even by non-technical people. The hosted web app will stay online until 30th November 2021 as a demo. After that, please contact me in case a reactivation is required.

I hope that you find the report interesting and useful. For any questions or queries, please contact me at ijazm@tcd.ie.

Thank you.

Bibliography

- [1] Erik Hjelmås, Boon Kee Low, Face Detection: A Survey, Computer Vision and Image Understanding, Volume 83, Issue 3, 2001, Pages 236-274,
- [2] Guangzheng Yang, Thomas S Huang; Human face detection in a complex background, Pattern Recognition, Volume 27, Issue 1, 1994, Pages 53-63,
- [3] M. Abdel-Mottaleb, and, A. Elgammal, Face detection in complex environments, in, Proceedings International Conference on Image Processing, 1999
- [4] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," IEEE Transactions on International Conference on Image Processing, vol. 1, pp. 900-903, 2002.
- [5] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. European Conference on Computer Vision 2014
- [6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition (CVPR) 2005.
- [7] P Viola, MJ Jones; Robust real-time face detection; International journal of computer vision, 2004
- [8] Shuo Yang, Ping Luo, Chen-Change Loy, Xiaoou Tang; WIDER FACE: A Face Detection Benchmark ;Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5525-5533
- [9] Vidit Jain and Erik Learned-Miller ; Fddb: A Benchmark for Face Detection in Unconstrained Settings ; Technical Report UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts, Amherst. 2010.

- [10] S. S. Farfade, M. J. Saberian, and L.-J. Li. Multi-view face detection using deep convolutional neural networks. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ACM) 2015
- [11] Joseph Redmon and Ali Farhadi ; YOLO9000: Better, Faster, Stronger; arXiv 2016
- [12] Joseph Redmon, Ali Farhadi ; YOLOv3: An Incremental Improvement; arXiv 2018
- [13] P. Adarsh, P. Rathi and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 687-694, doi: 10.1109/ICACCS48705.2020.9074315.
- [14] A. Womg, M. J. Shafiee, F. Li and B. Chwyl, "Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for Real-Time Embedded Object Detection," 2018 15th Conference on Computer and Robot Vision (CRV), 2018, pp. 95-101, doi: 10.1109/CRV.2018.00023
- [15] Zhang, S., Wen, L., Shi, H. et al. Single-Shot Scale-Aware Network for Real-Time Face Detection. *Int J Comput Vis* 127, 537–559 (2019).
- [16] Xu Tang, Daniel K. Du, Zeqiang He, Jingtuo Liu; PyramidBox: A Context-assisted Single Shot Face Detector; Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 797-813
- [17] Jia-Yi Chang and Yan-Feng Lu and Ya-Jun Liu and Bo Zhou and Hong Qiao ; Long-distance tiny face detection based on enhanced YOLOv3 for unmanned system ; arXiv 2020
- [18] M. Köstinger, P. Wohlhart, P. M. Roth and H. Bischof, "Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 2144-2151, doi: 10.1109/ICCVW.2011.6130513.

[19] Vahid Kazemi, Josephine Sullivan; One Millisecond Face Alignment with an Ensemble of Regression Trees ; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1867-1874

[20] Roberto Valle, Jose M. Buenaposada, Antonio Valdes, Luis Baumela; A Deeply-initialized Coarse-to-fine Ensemble of Regression Trees for Face Alignment; Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 585-601

[21] Marek Kowalski, Jacek Naruniec, Tomasz Trzcinski; Deep Alignment Network: A Convolutional Neural Network for Robust Face Alignment; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017, pp. 88-97

[22] J. Xiang and G. Zhu, "Joint Face Detection and Facial Expression Recognition with MTCNN," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 424-427

[23] M. Ma and J. Wang, "Multi-View Face Detection and Landmark Localization Based on MTCNN," 2018 Chinese Automation Congress (CAC), 2018, pp. 4200-4205, doi: 10.1109/CAC.2018.8623535.

[24] W. Zhao, R. Chellappa, A. Rosenfeld and P.J. Phillips, "Face recognition a literature survey," ACM Computing Surveys, vol. 35, no. 4, pp. 399-458, Dec 2003.

[25] B. Yang, J. Yan, Z. Lei and S. Z. Li, "Fine-grained evaluation on face detection in the wild," 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2015, pp. 1-7, doi: 10.1109/FG.2015.7163158.

[26] Guo Y., Zhang L., Hu Y., He X., Gao J. (2016) MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9907.

[27] Tianyue Zheng, Weihong Deng, Jiani Hu ; Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments ; Computer Vision and Pattern Recognition Arxiv 2017

[28] Jalal A., Tariq U. (2017) The LFW-Gender Dataset. In: Chen CS., Lu J., Ma KK. (eds) Computer Vision – ACCV 2016 Workshops. ACCV 2016. Lecture Notes in Computer Science, vol 10118. Springer, Cham.

[29] P. Terhörst, M. L. Tran, N. Damer, F. Kirchbuchner and A. Kuijper, "Comparison-Level Mitigation of Ethnic Bias in Face Recognition," 2020 8th International Workshop on Biometrics and Forensics (IWBF), 2020, pp. 1-6, doi: 10.1109/IWBF49977.2020.9107956.

[30] Acien A., Morales A., Vera-Rodriguez R., Bartolome I., Fierrez J. (2019) Measuring the Gender and Ethnicity Bias in Deep Models for Face Recognition. In: Vera-Rodriguez R., Fierrez J., Morales A. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2018. Lecture Notes in Computer Science, vol 11401.

[31] B. Chen, C. Chen and W. H. Hsu, "Face Recognition and Retrieval Using Cross-Age Reference Coding With Cross-Age Celebrity Dataset," in IEEE Transactions on Multimedia, vol. 17, no. 6, pp. 804-815, June 2015, doi: 10.1109/TMM.2015.2420374.

[32] Matthew Turk, Alex Pentland; Eigenfaces for Recognition. J Cogn Neurosci 1991; 3 (1): 71–86.

[33] M. S. Yang, N. Ahuja and D. Kriegman, "Face recognition using kernel eigenfaces," Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101), 2000, pp. 37-40 vol.1, doi: 10.1109/ICIP.2000.900886.

[34] H. Sahoolizadeh and Y. Aliyari Ghassabeh, "Face recognition using eigen-faces, fisher-faces and neural networks," 2008 7th IEEE International Conference on Cybernetic Intelligent Systems, 2008, pp. 1-6, doi: 10.1109/UKRICIS.2008.4798953.

[35] H Hanselmann, S Yan, H Ney ; Deep fisher faces ; BMVC, 2017 - bmva.org

- [36] Nabatchian, Amirhosein, "Human Face Recognition" (2011). Electronic Theses and Dissertations. 437.
- [37] W. S. Yambor. "Analysis of PCA-based and fisher discriminant-based image recognition algorithms." M.S. Thesis, Technical Report CS-00-103, Computer Science Department, Colorado State University, July 2000.
- [38] R.N. Belhumeur, J.P. Hespanha and D.J. Kriegman, "eigenfaces vs fisherfaces Recognition using class specific linear projection," IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, July 1997.
- [39] T. Ahonen, A. Hadid and M. Pietikainen. "Face recognition with local binary patterns," in Proc. ECCV, 2004, pp. 469–481.
- [40] C.H. Chan, J. Kittler and K. Messer. "Multi-scale local binary pattern histograms for face recognition," in International Conferences on Biometrics, 2007, pp. 809-818.
- [41] Zhang G., Huang X., Li S.Z., Wang Y., Wu X. (2004) Boosting Local Binary Pattern (LBP)-Based Face Recognition. In: Li S.Z., Lai J., Tan T., Feng G., Wang Y. (eds) Advances in Biometric Person Authentication. SINOBIOMETRICS 2004. Lecture Notes in Computer Science, vol 3338. Springer, Berlin, Heidelberg.
- [42] A. Petpon and S. Srisuk, "Face Recognition with Local Line Binary Pattern," 2009 Fifth International Conference on Image and Graphics, 2009, pp. 533-539, doi: 10.1109/ICIG.2009.123.
- [43] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao and Jianzhuang Liu, "Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor," in IEEE Transactions on Image Processing, vol. 19, no. 2, pp. 533-544, Feb. 2010, doi: 10.1109/TIP.2009.2035882.
- [44] P. Jonathon Phillips; Support Vector Machines Applied to Face Recognition ; Neural Information Processing Systems (NIPS) 1998

- [45] G. Guo, S. Z. Li and K. Chan, "Face recognition by support vector machines," IEEE International Conference on Automatic Face and Gesture Recognition, pp. 196-201, Mar 2000
- [46] I. Naseem, R. Togneri and M. Bennamoun, "Linear Regression for Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 11, pp. 2106-2112, Nov. 2010, doi: 10.1109/TPAMI.2010.128.
- [47] S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back, "Face recognition: a convolutional neural network approach," IEEE Transactions on Neural Network, vol. 8, no. 1, pp. 98-113, Jan 1997
- [48] Ralph Gross, Vladimir Brajovic ; An Image Preprocessing Algorithm for Illumination Invariant Face Recognition; AVBPA 2003: Audio- and Video-Based Biometric Person Authentication pp 10-18
- [49] P. Jonathon Phillips, Patrick J. Rauss, Sandor Z. Der ; FERET (Face Recognition Technology) Recognition Algorithm Development and Test Results, DARPA 1996
- [50] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPRW), Faces "in-the-wild" Workshop/Challenge, volume 4, 2017. 47
- [51] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In the European Conference on Computer Vision. Springer, 2016. 75
- [52] R.N. Belhumeur, J.P. Hespanha and D.J. Kriegman, "eigenfaces vs fisherfaces Recognition using class specific linear projection," IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, July 1997.
- [53] Harihara Santosh Dadi, Gopala Krishna Mohan Pillutla, Improved Face Recognition Rate Using HOG Features and SVM Classifier, IOSR Journal of Electronics and Communication Engineering 2016

- [54] XueMei Zhao; ChengBing Wei, A real-time face recognition system based on the improved LBPH algorithm, IEEE International Conference on Signal and Image Processing (ICSIP) 2017
- [55] J. Chen, V. M. Patel and R. Chellappa, "Unconstrained face verification using deep CNN features," 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1-9, doi: 10.1109/WACV.2016.7477557.
- [56] Florian Schroff, Dmitry Kalenichenko, James Philbin; FaceNet: A Unified Embedding for Face Recognition and Clustering; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823
- [57] E. Jose, G. M., M. T. P. Haridas and M. H. Supriya, "Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 608-613
- [58] T. Baltrušaitis, P. Robinson and L. Morency, "OpenFace: An open source facial behavior analysis toolkit," 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1-10
- [59] T. Baltrusaitis, A. Zadeh, Y. C. Lim and L. Morency, "OpenFace 2.0: Facial Behavior Analysis Toolkit," 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 2018, pp. 59-66
- [60] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf; DeepFace: Closing the Gap to Human-Level Performance in Face Verification ; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701-1708
- [61] I. Masi, Y. Wu, T. Hassner and P. Natarajan, "Deep Face Recognition: A Survey," 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018, pp. 471-478

- [62] Jiankang Deng, Yuxiang Zhou, Stefanos Zafeiriou; Marginal Loss for Deep Face Recognition; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017, pp. 60-68
- [63] Chen S., Liu Y., Gao X., Han Z. (2018) MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. In: Zhou J. et al. (eds) Biometric Recognition. CCBR 2018. Lecture Notes in Computer Science, vol 10996
- [64] Xianyang Li, Feng Wang, Qinghao Hu, Cong Leng; AirFace: Lightweight and Efficient Model for Face Recognition; Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 0-0
- [65] L. Guo, H. Bai and Y. Zhao, "A Lightweight and Robust Face Recognition Network on Noisy Condition," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPAASC), 2019, pp. 1964-1969
- [66] Hao Wang, Yitong Wang, Zheng Zhou ; CosFace: Large Margin Cosine Loss for Deep Face Recognition ; Tencent AI Lab, Computer Vision and Pattern Recognition (CVPR) 2018
- [67] Lin S., Tang J., Feng Z., Lai J. (2020) Deep Face Recognition Based on Penalty Cosface. In: Peng Y. et al. (eds) Pattern Recognition and Computer Vision. PRCV 2020. Lecture Notes in Computer Science, vol 12306.
- [68] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. arXiv preprint arXiv:1801.07698, 2018. i, 33, 46
- [69] Jiankang Deng, Stefanos Zafeririou; ArcFace for Disguised Face Recognition; Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 0-0
- [70] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 2018, pp. 67-74

[71] Massoli F.V., Amato G., Falchi F., Gennaro C., Vairo C. (2019) Improving Multi-scale Face Recognition Using VGGFace2. In: Cristani M., Prati A., Lanz O., Messelodi S., Sebe N. (eds) New Trends in Image Analysis and Processing – ICIAP 2019. ICIAP 2019. Lecture Notes in Computer Science, vol 1180