Real time dynamic facial recognition of subject at motion using angular image

by

Sharah Tasneem
20101186
Ramisa Yashfi Rahman
20101157
Ayman Mansur
20101432
MD. Meherab Hossain Nowshad
20301308

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
Jan 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sharah Tasneem
20101186

Ramisa Yashfi Rahman
20101157

Ayman Mansur
20101432

MD. Meherab Hossain Nowshad
20301308

Approval

The thesis/project titled “Real time dynamic facial recognition of subject at motion using angular image” submitted by

1. Sharah Tasneem(20101186)
2. Ramisa Yashfi Rahman(20101157)
3. Ayman Mansur(20101432)
4. MD. Meherab Hossain Nowshad(20301308)

As of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 22, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Co-ordinator:
(Chair)

Dr. Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In the developing world keeping track of violations or implementing a secured environment has become crucial. In order to address such issues dynamic facial recognition could be developed in such a way that it can facilitate and address all these issues. Dynamic facial recognition is a real time recognition of a subject while it is in motion. Different well known pre-trained models for facial recognition such as ResNet50, VGG19, VGG16, DenseNet169, Inceptionv3 and MobileNetv2 were customized according to the requirement of the dataset to bring about the highest accuracy. Before training the models, the process composed of several steps involving data acquisition which retrieved pictures from various angles of subject. To detect faces and create bounding boxes around the faces as well as marking facial landmarks such as eyes, nose and mouth MTCNN algorithm has been used. In order to compare, the test dataset was divided into two different types where one consisted of all the data and the other consisted of only the images with 120 degree deviation. This helped us to understand how feature extraction is an important factor for facial recognition as all the trained models provided improved and better results with the filtered dataset. Among all the models trained, it can be concluded that the best performing model for our custom dataset is VGG19.

Keywords: Deep learning; Facial Recognition; Angular Image; MTCNN; TensorRT; Computer vision; Dynamic; Angular deviation

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Importance of facial detection and recognition

In this modern digital age of today, facial detection, a branch of computer vision technology, has become a potent instrument with high demand and many uses [30]. Facial recognition is of utter importance in many fields nowadays. From enhanced security to personalised experiences, it is used in a wide number of fields. Many tasks have become way simpler, easier and better through the technology of facial recognition and detection which is a sub-field of computer vision [32]. This has been possible due to the rapid invention and development of machine learning and image processing algorithms. Facial recognition and detection are two subfields of computer vision which are related to each other. Facial detection is basically the method of recognizing one face from an image or a video. Different algorithms are used to discover and identify the face by analysing patterns of an image. Whereas facial recognition is a step beyond facial detection which aims to differentiate among the different features of the faces.

1.2 Dynamic facial recognition

Dynamic facial recognition is the detecting and recognizing of faces in a continuous environment using advanced computer vision algorithms and techniques. These algorithms are developed in such a manner that the face of a moving individual can be detected in spite of its facial appearance changing in every frame.

1.3 Challenges

Constraints such as facial expression, size, shape, light variations and misalignment of the face are common difficulties that arise during face recognition systems. Biometric methods such as eye retina and iris colour are also computationally demanding. Us humans are not perfect, our faces may have defects, capturing various images and creating mirrored images of the side view may result in numerous faults in the frontal image. Since automated real time face recognition is extremely sensitive to pose variations, precise and efficient rotation of the face or alignment of the image axis are required for better outcomes [11]. Changes in lighting condition is also a major factor and can be resolved using robust machine learning techniques.

1.4 Problem Statement

Over time, crime on the road, and in public places has been increasing exponentially. For instance, the city of Arizona has seen a rise in crime rate in the last 5 years. Assaults and drug crime have risen some folds. The crime rate in 2016, was somewhere near 400 cases however in just five years it has risen to roughly 800 cases [31]. This is not only true for the city of Arizona, but also true for the entire world. Especially in developed and developing countries. These countries lack the proper infrastructure to prevent or solve crimes.

In Bangladesh for example, according to [19] there had been 3089 cases of reported crime in Dhaka which is the capital city of Bangladesh In 2019. And 2396 cases in Chittagong. However, the approach to solving these reported crimes has been fairly primitive. Solving crime or just taking action takes a lot of time and hence remains mostly unsolved. And even if it's solved it requires a prolonged waiting time period. This is mainly because identifying criminals in these settings can prove to be difficult as recognizing faces might not always be easy. Moreover, if the setting is in a crowded area, it can be even more difficult. Identifying a single person in order to ensure safety or in any other circumstances can be hard, time-consuming and difficult. Detecting a subject's facial features in a large crowd is a tedious process as these places are busy and crowded especially if the object is in motion.

Not only that surveillance at universities, workplace settings and school might require extra care to begin with which also is capable of helping in keeping track of individuals for example, keeping automated attendance. As this process of keeping track of workers or students can be quite time-consuming, and tedious. Again, there is also a need for contactless authentication which can ensure public health safety at hospitals as well. The importance of which was realised after the pandemic of Covid-19. Therefore the lack of proper identification system where the subject is at motion is an unexplored field.

1.5 Research Objective

Our study aims to address and achieve such a model that will be able to recognize face from a dynamic motion. In order to achieve this, we intend to have the following research objectives:

- Have a clear understanding of the different algorithms and techniques used for facial detection, recognition to be able to bring forward an efficient result for the custom dataset.
- Look into already existing research of detecting faces in dynamic motion to have an idea which will help us build the model in a better way.
- Learn face detection algorithm MTCNN.
- Study and analyse the architecture of facial recognition CNN models such as VGG19, Resnet50, DenseNet, etc.
- Try to implement TensorRT to reduce inference time for real-time.

Chapter 2

Literature Review

The paper [24] combined the YOLO algorithm with the VGG16 pretrained CNN model to propose better face detection systems. Their goal was to enhance face detection in multiple face positions, skin colors, and various lighting conditions for real-time live video. For preprocessing, they used the FDDB dataset (Face Detection Dataset and Benchmark) that had over 28000 images, but they used 5000 images for their model. Then they created a ground-truth dataset that had images and labels for each image. Moreover, to solve the issue of overfitting, they used data augmentation, which provides for additional variation in the training data without increasing the number of labeled training samples. Furthermore, for feature extraction, they used the VGG16 model and removed the unnecessary layers. The outputs generated in the pre-trained VGG16 model were further combined with the YOLOv2 algorithm for enhancement. The YOLO algorithm is a single-stage object detection system that is faster and has high accuracy in detection systems. In conclusion, their proposed model of YOLOv2 and the VGG16 network model showed an accuracy of 93%, which is comparatively higher than other models with YOLOv2. Although this model showed higher accuracy, it requires the use of high-end graphics cards for better computational results.

In this day and age, face recognition is tremendously popular in the computer vision field. There are many models that exist, but the majority of them are sophisticated with deep layers and have to be trained with a very large dataset that is computationally intensive. To solve this issue, [18] a light-CNN was proposed which had an inclusion of a modified VGG16 model in order to recognize faces with a limited dataset. The VGG16 architecture is composed of deep layers with multiple convolutional layers with a number of kernels, followed by max pooling. So, to minimize the complexity of the VGG16 model, they removed a few layers and filters, and the architecture became more concise. Their dataset had about 7,500 images, which they divided into a 5:2 ratio for training and validation. They performed 50 epochs for training using the stochastic gradient descent optimizer and then compared the results with the baseline VGG16 architecture along with theirs, and the accuracy for the baseline architecture is 77.8% and their proposed model is 94.4%. This demonstrates that the light-CNN model outperforms other models in terms of accuracy and the time required for training the model when the dataset is constrained and the number of divisions is small.

The authors here [13] tried to implement a compressed version of convolutional neural network. It has been implemented for image recognition of a dataset which is small model size and has a shorter training time. In order to address this problem, residual squeeze VGG16 has been used, which is a compressed convoluted neural network. This model when compared to VGG16 is 88% smaller in size and 24% faster while training hence consists of only the best part of VGG16. On comparing this model with SqueezeNet, this model is easily adaptable. This model has 12 firing modules and 4 convolutional layers. This paper has approached to connect VGG16, residual learning and squeeze technique in forming a model which is smaller and faster by creating effective residual connection in fire module. On comparing the top-1 validation of VGG16 and Residual Squeeze VGG16, an accuracy of 54% and 51.68% was achieved respectively and for top-5 validation an accuracy of 84.3% and 82.04% however VGG16 took 88 hours whereas the proposed model took only 67 hours. To conclude, this model is able to address speed and size.

The study [27] has tried to address the uncontrollable situations caused by varying light intensity and angle while capturing content of samples of spatial and angular information by using a framework of double deep spatio angular learning. The proposed model is visual geometry group-19 (VGG-19). In order to capture the content, detection of light ray and light intensity is done by a plenoptic camera and a VGG-19 model along with a double deep system. Firstly, an input image is taken in and is pre-processed inorder to reduce noise. Then it is passed on through a spatio-angular face extractor,and subsequently passed through a VGG-19 and CNN model pair to study and analyze along with trained dataset to generate an output. Moreover, a comparison between VGG-16 and VGG-19 was made in order to compare between accuracy, sensitivity and specificity. VGG-19 had 15% more accuracy. 10% more sensitivity and 6% more specificity. Lastly it is concluded that this model is highly false detecting.

This paper [17] researches face recognition using correlation recognition algorithms based on the facial features extracted from the images. The main challenge here is to choose the most appropriate feature extraction method and matching the strategy used. According to the paper, the available extraction method of face recognition can be divided into categories where one of them is based on geometric features of faces which is a very early technique hence it requires very high quality, straight and vivid face images. The next method is based on statistical characteristics as the name suggests this works with statistical techniques to pinpoint various facial features and then treats the image as a random vector. Lastly, the paper mentions the extraction method based on neural networks which has positive results but a huge amount of data is required which results in increased computation cost as well. Due to all the problems in these methods, the paper proposes deep learning method VGG-16 to extract features, PCA (Principal component analysis) to prevent irrelevant features from taking part in this procedure and finally SVM algorithm which is a classifier to predict the sample. VGG-16 uses Euclidean distance classification method for face recognition. Here, the authors have used CelebA and LFW face image datasets where they have used the Dlib algorithm to preprocess the images. So basically, the procedure entails feature extraction with the VGG-16 network, data splitting into training and test sets, minimizing dimensionality with PCA, normalization of the data, and the training of an SVM-based face recognition model that is then used to identify faces in the test set. This procedure was able to provide a 97.47% accuracy. For future work, in order to improve operational efficiency, the authors mention to concentrate on how to optimize the network structure and develop a custom feature extraction technique.

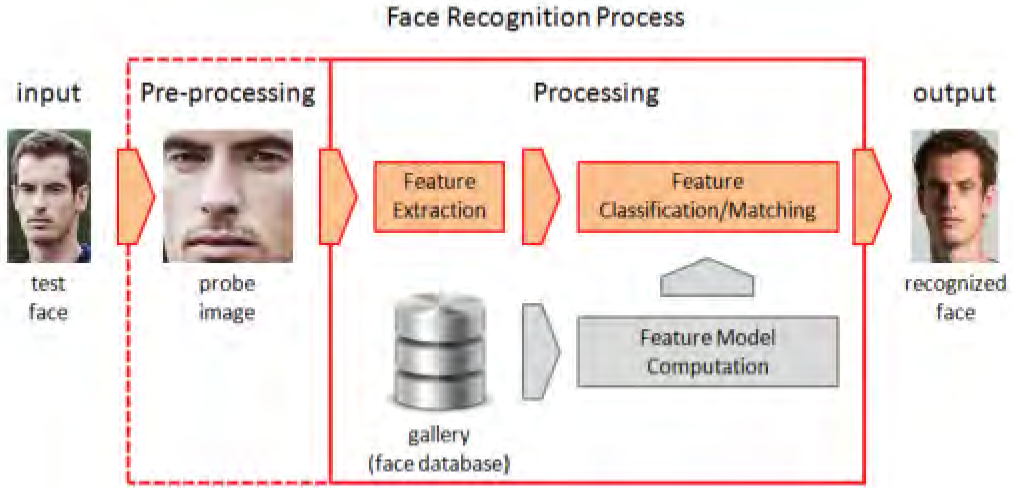


Figure 2.1: Schematic representation of the automatic facial recognition [11]

Given above is an automated face recognition system that takes an input image and uses robust machine learning techniques to pre-process the image which includes feature extraction, feature classification (compares from the database), and outputs the final recognized face.

Here, the paper [7] focuses on FRAD (Face Recognition at a Distance) which aims to recognize subjects from a distance even without knowing about the subject. They start by accurately identifying and following people in fixed camera surveillance footage (WFOV). Then by using a Kalman filter to track subjects of the real world metric coordinate system in the ground plane, it is possible to estimate where an automatically operated PTZ camera (NFOV) will be able to record the subject's facial image. Many subjects could be viewed at once using the person tracking system. Next, using a commercial face recognizer, the images that were collected were processed. With the help of this biometric surveillance system, targets can be tracked across distances of 25m to 50 m and recognized subject at 20 m radius. In the presence of a networked system of nodes, the subjects can be correctly labeled by any of the nodes within an area once they have been identified. All in all, to efficiently monitor and identify persons in a dynamic environment, this system integrates cutting-edge camera technology, tracking algorithms, scoring processes, and facial recognition software. While assuring precise and effective image capture and recognition, it prioritizes participants depending on their behavior and prior interactions. The authors find that the average range for detection and capturing of face is roughly twice as large as facial recognition, which has been found by calculating the mean and standard deviation value for the detection. This is to be expected given that the current state of facial recognition technology is limited by the requirement for high quality facial photos. Moreover, higher ranges make it more challenging for face matching software to function properly because the acquired facial images are smaller overall and more susceptible to motion blur from the moving subject.

Mainly with Convolutional Neural Network (CNN), the researchers of this paper [14] worked on facial expression recognition. This method can be used in various applications, such as security, human-computer interaction and behaviour understanding.CNNs is a powerful tool for image-based classification tasks. Three approaches to image appearances have been discussed as preprocessing techniques which are Region of Interest(ROI), Difference, and Local Binary pattern (LBP) images. The convolutional layers, pooling layers, and fully linked layers used in the CNN architecture are described. The study shows experimental outcomes for CNN training on three freely accessible face databases for facial emotion recognition: JAFFE, FACES, and CK+. Inception-v3, VGG16, VGG19, and VGG-Face are the pre-trained deep CNN models that are evaluated in this study to examine the use of transfer learning. The results showed that VGG-Face, a model pre-trained specifically for face recognition, generally outperforms Inception-v3 and VGG16/19, which are pre-trained for object recognition. But mostly, VGG16 performs similarly or even slightly better than VGG-Face. To conclude, the use of deep learning models, especially pre-trained CNNs can be a competent way for facial recognition. The practicality and efficiency of transfer learning in FER tasks and human-machine interaction.

This paper [25] is about a brief study of exploring the use of Convolutional Neural Networks (CNNs) which can be used for facial recognition with the purpose of user security. The paper demonstrates the need to use facial biometrics as a complementary security mea sure besides username, password and fingerprint biometrics. Eight prominent CNN models have been implemented including Alexnet, Xception, Inception v2, Inception v3, ResNet50, ResNet101, VGG16, and VGG19 for effectiveness in recognizing facial images. Both VGG16 and VGG19 showed the highest level of image recognition accuracy and F1-Score among the tested CNN models. The research utilizes the dataset of Labeled Faces in the Wild (LFW) and employs a Python environment with the Keras library. The tested CNN models did not meet the required accuracy level for immediate response, but future research with larger datasets, additional classification techniques, and evaluation models could enhance the security system's strength and reliability.

The study [22] here is based on the YOLOv4 face detector that detects real-time masked faces and unmasked faces scenarios and classified using an effective CNN archi tecture. Their work is mainly focused on men and women from iran with different clothing such as hijabs, masks, etc and most importantly majority of the dataset performs abysmally in the real world usage. They took a combined total of about 8000 images from MAFA and WIDER FACE respectively. Moreover they collected 1500 images of Iranian people too. While labeling images by bounding box they stored 5 parameters. The first parameter determines if the person is masked or unmasked and the rest of them are the coordinates of the bounding box. Data augmentation has been performed using random perspective transformations, brightness alteration and inclusion of Gaussian noise that increased the dataset to 25000 im ages. In their proposed method at first the YOLOv4 is used to detect objects , the masked and unmasked are distinguished as two separate classes. In their second proposed method they combined both masked and unmasked into one class. All the cropped images were fed into a fast effective classifier. Thus, for the second classifier an accuracy of 99.5% was achieved.

As of today's date, AI development has gone to another level, as mentioned in [28] two of the most common examples of this progress are self-driven automobiles and self-service supermarkets. The need to subject AI is because computer vision is hugely connected to AI which focuses on replicating human vision and aims on capturing images just as the human eye does. By using a deep learning approach and training the data using the CNN approach, the authors of [28] developed the face recognition system with an accuracy of 91.7% given some factors such as lighting and distance of the face are kept in mind.

In spite of being in this modern era, social security isn't the best and in fact, is deteriorating. Keeping this in mind, to provide an enhanced solution, [8] focuses on building a four-axis dual-CCD system that will be able to recognize faces in real-time 3D. Unlike the common approaches, the authors here used a bionic visual system which is a client-server-based system along with the added benefit of reduced hardware cost. A geometric relationship is applied to detect the faces and then a fuzzy classification is used for identity recognition. Image pre-processing is a crucial step of facial recognition. Including movement tracking and the use of low-pass filters images are smoothened for better recognition.

The study [29] presents multi-angle facial recognition in video sequences. They use a mechanism for face localization which is, 2D facial feature point estimation and a utilization of GAN technique to create a frontal face image. The authors point out two limitations in traditional face recognition methods. They used Dlib (a machine learning library based on C++) for frontal face images resulting in 99.38% validity on the image dataset. No matter how precise the results are, the library has its limitations due to large pose faces. To improve this, they used TP-GAN which enhances the ac curacy by removing large pose faces while maintaining the general face structure composition. Moreover, CNN model has been used to train a new classifier and im provements were made with MULTI-PIE for a better facial output. Several images from different angles were collected to train the model. Therefore, the final outcome achieved was better than the traditional facial recognition systems.

K. Anderson [4] discusses about a face tracking system that can recognize faces in different lighting conditions. It assigns a single face probability using some techniques to each subject in the scene which are fusing picture measurements, eye tracking data, and a modified ratio template technique. Besides, an optimal flow diagram is used to detect movement. The multi-channel gradient model (MCGM) functions in three dimensions, reconstructing the dense velocity field of the image at all locations. This paper offers useful knowledge about the formation of a face-tracking system and its future uses.

A scientific paper that discusses the use of 3D dynamic sequence of facial scans, using a fully automated approach for facial expression which is reasonably simple, has a low computational cost, and has time applications [9]. As a result, each 3D frame in the sequence is analyzed individually and deviations along the time dimension are acquired and categorized with Hidden Markov Models (HMMs). The technique concentrates on specifying the distances between facial landmarks and obtaining local features at those points. The local descriptor at facial landmarks is used to check if there's any deformation caused by the surrounding people's facial expressions. The 3D surface around a given landmark can be defined by defining an invariant local Reference Frame RF one or more geometric measurements in accordance with the local coordinates, and using a local surface descriptor. The geometric measurements take into account each and every point on the support. The author of [9], also discusses the use of a descriptor that is similar to a shape context to explain local deformation at a few important facial landmarks. To account for the reciprocal movements of the landmarks, the Euclidean distance measured in 3D space between them is used as a relational descriptor.

Demonstration of a face recognition system for real time applications with the help of Support Vector Machines(SVM), FaceNet and Multi-Task Cascaded Convolutional Neural Network(MTCNN) has been discussed in this paper [26]. The main focus of this work was to get high recognition rates with low training time. The combination of FaceNet and Support Vector Machines has been used there where FaceNet helps to extract face features and Support Vector Machines for classification. MTCNN performs 3 stages to detect bounding boxes of faces with 5 point face landmarks. The system achieved high accuracy with 99.85% for straight faces, slight deviations and faces with the head lifted up. With the ability to handle variation in angle, tilt the suggested method is also effective in real-time face recognition.

The paper [1] introduces a facial expression recognition system based on Facial Motion Graphs (FGM) and Continuous Dynamic Programming(CDP). The main concern of this system is to achieve the accuracy of approaches in recognizing facial expressions. In order to recognize facial expressions, the system applies Continuous Dynamic Programming(CDP) which helps to calculate the distance between two FMG sequences in considering optimal correspondence. To get each edge in the FMG they used the weighted local distance method and the weights are calculated based on the separability of expressions with the help of edge waveforms. To show the result of the experiments a video database has been conducted which includes four types of facial expression (Anger, Dislike, Happy, Surprise). In conclusion, that system is able to accurately recognize facial recognition and spot them in image sequences which were found out from the experiment of a video database.

Here the paper [20] proposes a model with a combination of MTCNN and VGGnet. MTCNN has played the role of detecting the image and VGG helps to extract the other complex features from the face. The MTCNN provides better results in terms of robustness to light, angle and facial expression changes. The model is trained and evaluated on datasets like CASIA-WebFace and LFW which demonstrates really competitive performance compared to other deep learning algorithms.. The method provides a balance between accuracy with approximately 98.53% and computational efficiency that makes it suitable for performing in real-time applications.

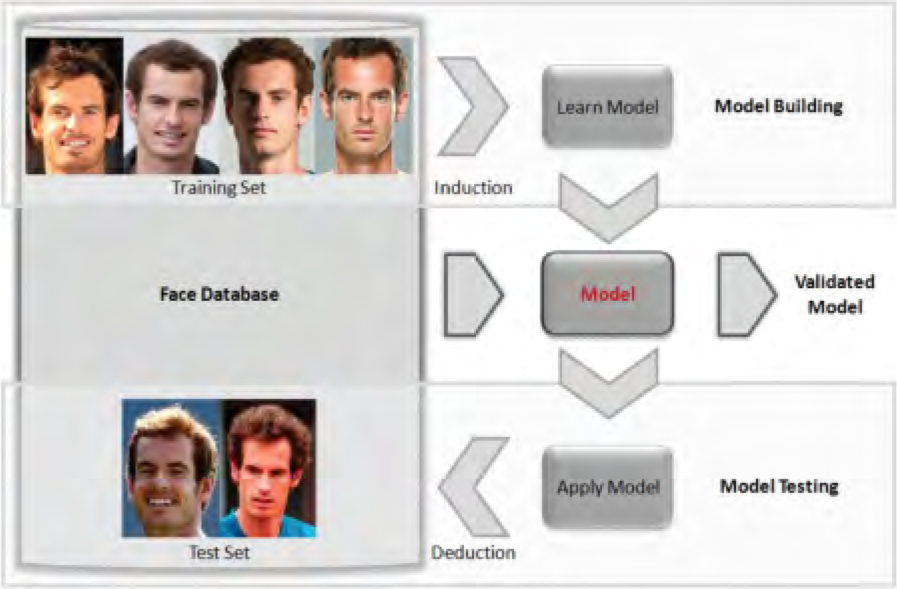


Figure 2.2: Overview of the model computation [11]

Multiple camera recordings for a conference/seminar are expensive. The author of [3] proposed face tracking through a single omnidirectional camera for an office meeting capturing 360° situated in the middle of the conference table. They captured the photographs using a typical video camera which was equipped with a hyperbolic mirror that enabled them to capture almost 360° of the field view. They transformed the 360° to a normal perspective image by using sub-pixel anti-aliasing methods.They distorted the image by bringing on to plane from again a perspective cylindrical part, and still it twists and curls itself into that boundary direction. All these processing mentioned has to be done before the face tracking. They also used two skin detection and segmentation methods. Gaussian mixture model(GMM) was used to transform RGB-colour intensities to rg-Chroma space to balance the skin colour in different lighting conditions. Next, a Global Skin Colour model is used for a more robust outcome which generates a skin colour probability. In addition, they applied Neural Network techniques to look for faces using skin colour. Lastly, they used a Particle Filter based tracking system to detect the number of points in faces. Although, this system of using a single omnidirectional camera is computationally expensive but rather efficient in most cases.

In this day and age, technology is expanding rapidly making facial recognition systems more effective yet complex. [12] presents a simple real-time human face detection framework. The authors' motivation was to build an efficient face recognition system to detect criminals from databases like NADRA (Pakistani agency). Their proposed system requires necessary pre-processing for detection. Viola Jones algorithm is used for human face detection and outliers are removed by MSAC algorithm. The images from the dataset are pre-processed for better noise reduction, and a connected neighbourhood algorithm is used. An adjacent sum of pixels (x,y) is used for row summation. Furthermore, the image is cropped so that more features can be extracted. Speeded Up Robust Features (SURF) has also been used for image recognition. The Hessian matrix has been integrated for interest point detection and wavelet responses are used for interest point description. 16 sub-squares are created from each square image, and each sub-square is then divided into 4 further squares.Then the Haar wavelet responses are calculated, resulting in a final feature vector length of 64. This system works best indoors and the result of this proposal showed that using an old spec computer takes an average of only 1.96 seconds for comparison between 2 images which manifests that using better hardware would result in superior response time which is beneficial for real-time detection.

Multidirectional face recognition has always been challenging. [10] depicts how generating a face recognition system from a side-angled image helps in surveillance applications. The motive of this study was to compare the results of a side-view image with the target image that is kept in the dataset. The author uses various methods such as Viola Jones's skin detection technique for feature extraction. It develops a 2D side view image and copies the best side to create a mirror image of the face as a 2D mug shot face. For image-based techniques, the target image may not match the dataset image angle so they require pre-processing of the image such as scaling and rotation. Lastly, biometric methods can also be utilised in a side view image such as biometric features like detecting eye retina and irises [5]. For face recognition, the author proposed various steps that extract the feature, create a split face and generate a mirror image and at last they enhance the features of the face for a better outcome. Therefore, this process is useful for 2D images and results in an increase in success rate to match with reference images for human recognition from the side view face image.

The paper [33] focuses on child face recognition using Convolutional Neural Networks(CNNs) which could play an important role in finding missing children, ensuring school safety. The study evaluates the performance by using VGG16, ResNet50 and MobileFaceNet on a child face dataset where MobileFace secured his place on top by achieving accuracy of 99.75%. The study discusses the difficulties in recognizing faces in children because of morphological changes and emphasizes the use of deep learning for automating feature extraction. Metrics including recall, specificity, precision, false acceptance rate, false rejection rate are also used to assess the suggested models.

Holistic face recognition uses global information from faces to perform face recognition [6]. This approach can result in better execution than feature based approaches but its performance declines when there is dissimilarity in facial expressions or poses [2].The authors of [6] use Scale Invariant Feature Transform (SIFT) for face recognition. The propose two innovation methods. The first, which is known as VSIFT, focuses on the volume structure of an object and eliminates those key points that could not be trusted. Following this, a methodology known as the Partial-Descriptor-SIFT (PDSIFT) was used towards the identification of important points when large scales are applied at face boundary levels. They took the images from the AR dataset and converted them to grey-scale and cropped them to 60 x 85. A total of 75 persons' images were taken in consideration for the training test. They compared the results with traditional feature based approaches and concluded that PDSIFT achieves better results than that of original SIFT approaches.

Chapter 3

Working Plan

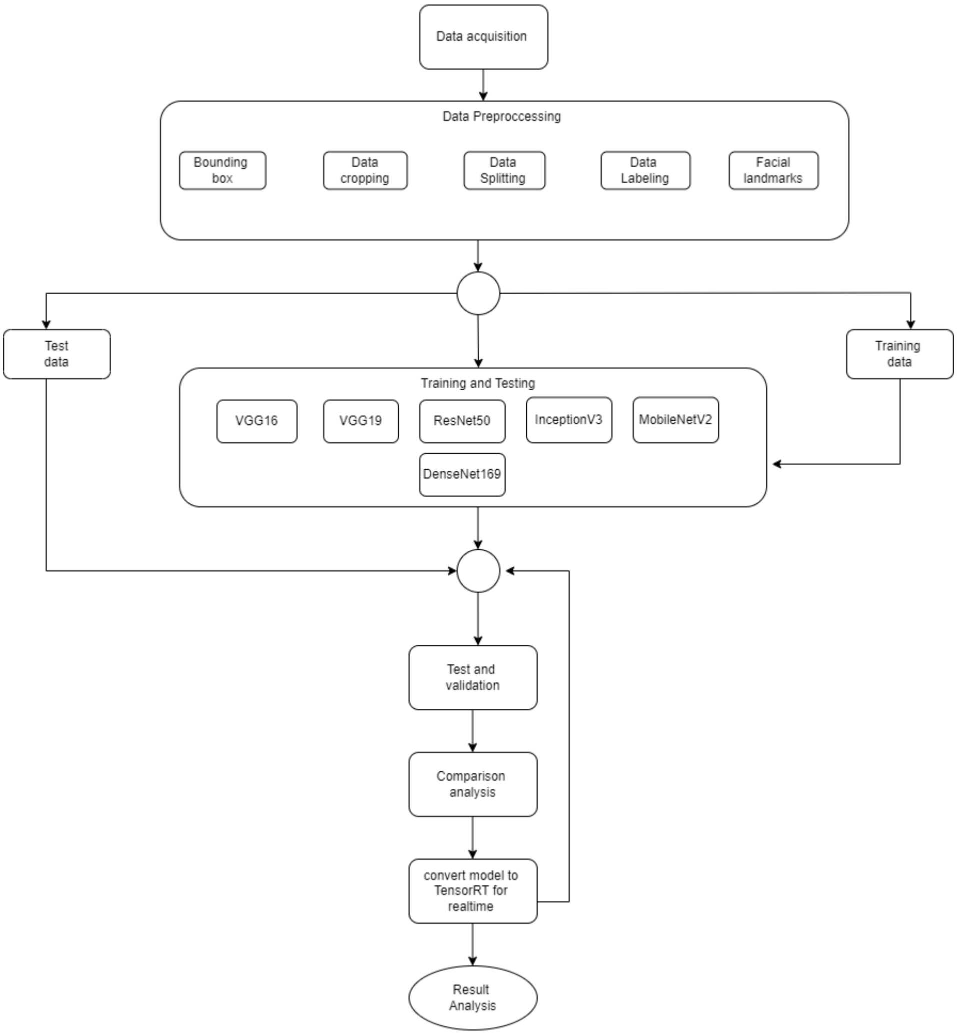


Figure 3.1: Tentative work flow

To attain our end results successfully a fair start was achieved by reviewing available papers that coincides with our topic. The main target was to find relevant topics involving facial recognition techniques, expression analysis and facial feature analysis. The existing models VGG19, MobileNetv2, ResNet50, DenseNet169, InceptionV3 and VGG16 were planned to be implemented. Then analyzation of each model's performance on the dataset will be recorded in order to compare and evaluate the results. In order to test the trained models, two datasets will be prepared from one core test dataset. Finally, in order to reduce inference time for the real-time implementation, we aim to convert our model to TensorRT.

Chapter 4

Description of models & Hardware configuration

4.1 Model Descriptions

4.1.1 VGG-19

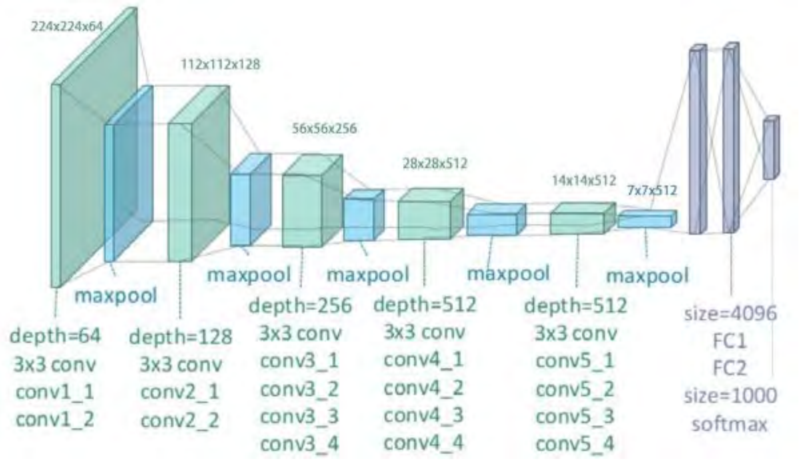


Figure 4.1: VGG-19 architecture [16]

Visual Geometry Group 19 or VGG-19 is a deep convolutional neural network architecture. It has been designed for image processing and classification which can be made responsible for different recognition tasks. It comprises of 19 layers which can be subdivided into 16 convolutional layers and 3 fully connected layers. These 19 layers are uniform and well structured which helps in better image recognition. To begin with, the input layer of VGG-19 takes an input as a RGB image of size 224x224 pixels. This is further connected with 16 convolutional layers each followed by a rectified linear unit or ReLU activation function. The convolutional layer has a 3x3 receptive field. To retain spatial dimensions, the convolutional layers feature modest 3x3 receptive fields with a stride of 1 and zero-padding. Moreover , after every two convolutional layers there is a max-pooling layer. Max-pooling is used to reduce image size without disregarding the important information. These 16 convoluted layers are further connected to 3 fully connected layers. Each neuron transformers the input vector through the weight matrix and passes down information. The third layer of this fully connected layer is the output layer which consists o a number of classes. After the output layer follows the softmax, this is the final layer of VGG-19. This model is useful in its simplicity and uniformity. This makes its implementation and work much easier. Moreover, as this is a pre-trained model, it is a host to a large amount of dataset and is able to learn from the dataset and analyze.

Advantages of VGG19

Compared to other models, VGG19 achieved more popularity and the main purpose of us choosing this model for its simplicity and good performance. VGG19 has a straightforward and uniform architecture which makes it easy to understand and implement. Again, VGG19 is also capable of transfer learning for its simplicity. Moreover, this model showed good performance on image classification tasks including work of large and diverse dataset also.

4.1.2 DenseNet169

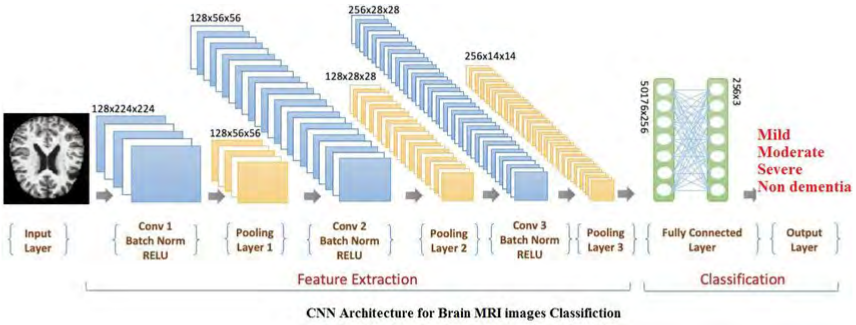


Figure 4.2: DenseNet169 architecture [15]

DenseNet169 or Densely connected convolutional network is a deep convolutional neural network architecture. Desnet169 comprises of densely connected blocks where each layer receives direct input from the preceding layers. This architecture promotes feature reuse and enables efficient training of the deep network. The input layer of DenseNet169 is responsible for taking an RGB image as input with a fixed size of 224X224 pixels. This model has networks which begin with an initial convolutional layer which extracts basic features from the input image. This layer also comprises a max-pooling layer, which is used to reduce the spatial dimension of the feature map. This is followed by a dense block which is the core building block of the architecture. Each of this dense block comprises multiple densely connected layers. Each of these layers receives feature maps from all preceding layers as an input which is added to the output generated by itself. This dense connection encourages feature reuse and facilitates learning ability of the model. Moreover these blocks consist of batch normalization followed by a rectified linear unit or ReLU. between each dense block there is a transition block. These blocks include a batch normalization layer which is used to reduce the number of channels. This is a 1x1 convolutional layer which helps in controlling the growth of parameters and computational costs. At the end of the network, global average pooling is applied to reduce the spatial dimensions of the feature maps to a1x1 size. Finally a fully connected layer and a softmax activation is implemented. This layer is responsible for mapping the output of the global average pooling to a desired number of classes. This model successfully gets rid of the vanishing gradient problem, and facilitates the training of the model. Moreover, due to its dense connectivity, it is able to achieve better performance.

Advantages of DenseNet169

As the name suggests DenseNet169 provides dense connectivity, where each layer receives input from all preceding layers which helps it enable effective learning and model performance. Dense block allows this model to gain diverse and effective features at different levels of abstraction. Again, DenseNet architectures are parameter-efficient which made the model more computational and easier to train.

4.1.3 MobileNetV2

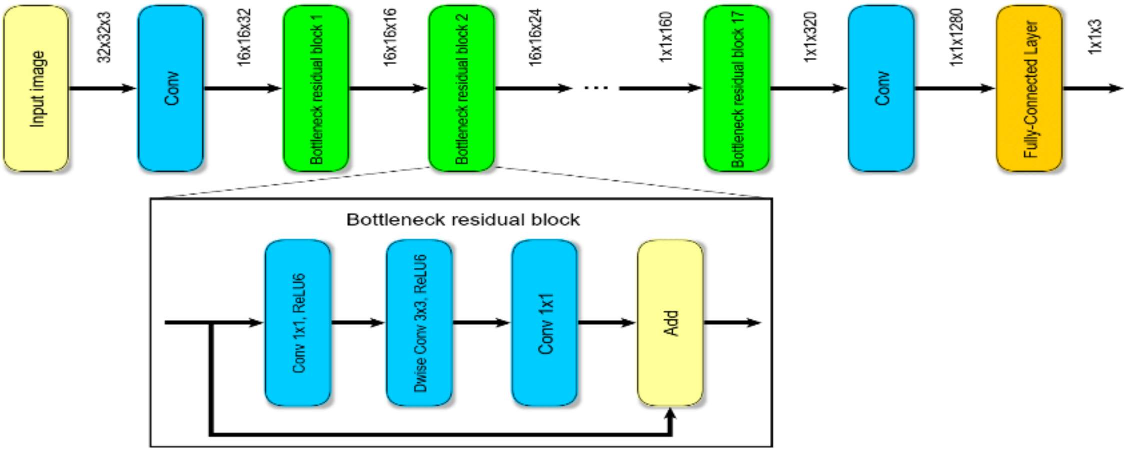


Figure 4.3: MobileNetv2 [21]

MobileNetV2 is a convolutional neural network (CNN) architecture which has been optimized for image categorization specifically on mobile and edge devices. It has a standard 3x3 convolutional layer with a stride of 2. Then the model has several bottleneck blocks consisting of multiple sublayers. There is a 1x1 Expansion Layer which works in increasing the number of channels. There is also a Depthwise Separable Convolution. Depthwise convolution and pointwise convolution are the two fundamental parts of the depthwise separable convolution. This convolutional process in general minimizes the amount of parameters and calculations, making the model more less complex. There is also a 1x1 Projection Layer to reduce the dimensions. A shortcut connection between the block's input and the Convolution Layer's output when they have the same dimension. Moreover this helps in the process of data reserving and training process. By utilizing shortcut connections and linear bottlenecks, the architecture presents inverted residuals. This architecture preserves efficiency while assisting in better capturing non-linearity. In the Squeeze and Excitation block, the consequent update of the feature maps based on their importance results in capturing the dependencies of the channels.Fully connected layers are avoided by the MobileNetV2 model and Global Average Pooling (GAP) is used instead to reduce spatial dimensions. This process makes the model simpler and the number of parameters are minimized.To control the spatial dimensions and maintain the efficiency, strides and padding are selected and applied calculatively throughout the processing of the network. Batch normalization and ReLU activation functions are applied after each convolutional layer to improve the training and to introduce non-linearity.

Advantages of MobileNetV2

MobileNetv2 is a lightweight convolutional neural network architecture. This model provides several advantages over other models in terms of efficiency and size. MobileNetv2 is designed to be computationally efficient. It has a small model size compared to other models which makes it suited for deployment on some resource constrained devices. MobileNetV2 architecture allows use of fewer parameters which helps reduce the memory footprint hence limits RAM usage. MobileNetV2 also has a highly efficient design, thus its power usage during the training and inference phases is lower.

4.1.4 VGG16

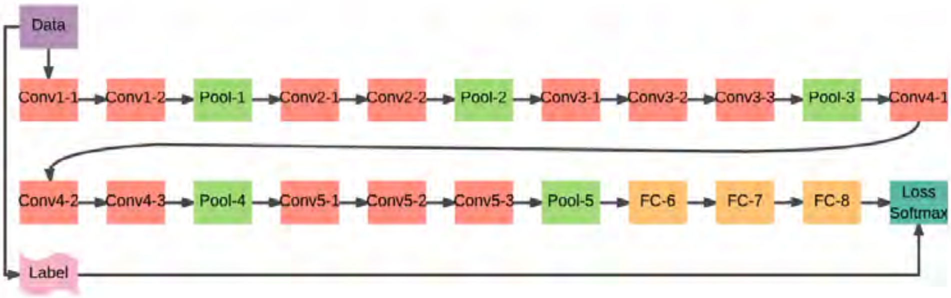


Figure 4.4: VGG-16 architecture [13]

VGG16 is a deep convolutional neural network architecture specifically designed for the work of image classifications. The VGG16 model takes an image as input, a RGB image generally 224x224 pixels. The architecture of the VGG16 model is very simple, containing several convolution layers backed by multiple max pooling layers. The convolution layers contain 3x3 filters including a stride of 1. The padding value is being kept constant to maintain the spatial resolution due to the convolution process. The initial convolutional layers extract low level features such as the simple patterns, texture and edges. Simple patterns, textures, and edges are examples of low-level features that are extracted by the first convolutional layers. The max pooling layers of this architecture reduces the spatial dimensions and provide a translation invariance by following the convolutional layers. The convolutional blocks gradually increase the depth of the network as they are stacked. High level information and intricate patterns are detected and worked on by the succeeding convolutional blocks as the network becomes deeper.Following the convolutional layers, there are three fully connected layers. First two of the connected layers have 4096 channels each whereas the third fully connected layer which is the output layer has the same number of channels compared to that of the classes of the classification task. The high level features are combined by the fully connected layers at the end of the network to make predictions about the class of the input image. After every convolutional layer and fully connected layer there are Rectifier Linear Units (ReLU) working as activation functions. In the output layer, softmax activation is implemented to generate probability scores and perform the multiclass classification.

Advantages of VGG16

VGG16 is one of the models belonging to a family called Visual Geometry Group, whose members are known for their simplicity and effectiveness. This simplicity allows it to be easily understood and applied, making it ideal for educational purposes. More specifically, pre-trained versions of VGG16 on large datasets such as ImageNet are available to use the learned features in various computer vision tasks. The characteristics of the deep architecture allow the model to learn delicate features represented by input data, which makes it suitable for tasks focused on describing intricate details.

4.1.5 Resnet-50

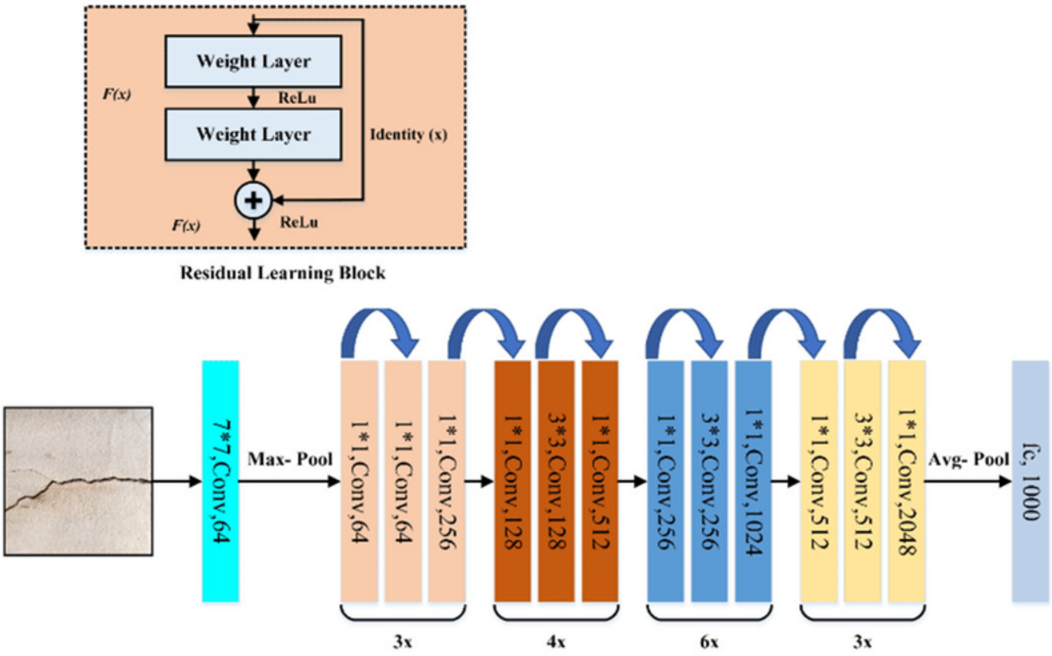


Figure 4.5: Resnet-50 architecture [23]

Residual network model or Resnet-50 is a deep convolutional neural network architecture consisting of 50 layers. It has an innovative way of using residual learning which addresses the vanishing gradient problem. The model has been constructed with certain layers, the input layer takes the input image as an RGB image with fixed size of 224x224 pixels. The 50 layers of ResNet-50 are split up into 5 blocks each containing a set of residual blocks. The convolutional layers of the network perform convolutions on the input image. Then this layer further performs max pooling the result of which is passed down to the residual layer. In the residual layer are 2 convolutional layers followed by a normalization layer. The core of ResNet is the use of this residual layer. Each residual block contains 2 distinct paths namely, identity path and Residual path. The identity path is a shortcut which connects the input to the output whereas the residual path has several stacked convolutional layers with batch normalization and ReLU activation. The final layer of the network is a fully connected layer which takes the output of the last residual layer and moves it to the next layer which is the output layer.

Advantages of Resnet-50

The main advantage of ResNet50 is the Residual Learning, which always helps in tackling the challenge of training very deep neural networks. This model also introduced the concept of 'state-of-the-art' performance in the field of image classification which takes this model to another level by achieving high accuracy rates in comparison to other previous architectures.

4.1.6 InceptionV3

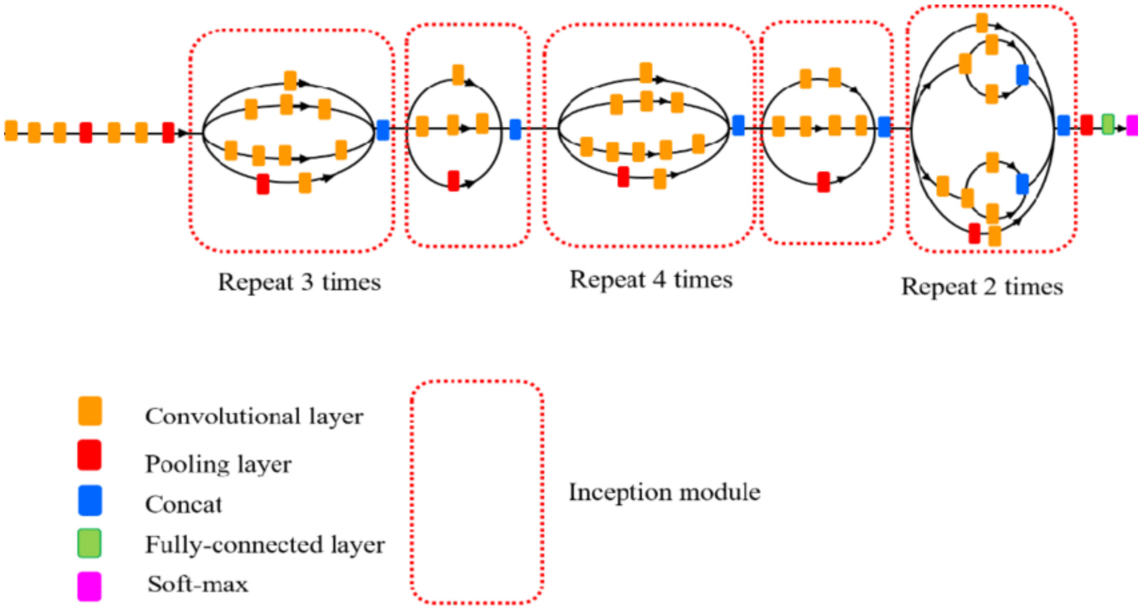


Figure 4.6: InceptionV3 architecture [15]

InceptionV3 is a convolutional neural network architecture. One of the main features of InceptionV3 is the use of its module which allows the network to capture multi scale features. To achieve this, InceptionV3 uses a combination of filters of different sizes within the same layer. The input layer of this model takes an RGB image of standard size of 299x299 pixels. This input is carried into the initial convolutional layer. This layer is responsible for extracting basic features and reducing spatial dimension, which could be achieved by a max-pooling layer. The core building block of the inceptionV3 layer are the inception modules. Each of these modules consists of multiple parallel convolutional and pooling operations of different sizes. These parallel operations are responsible for capturing features at different scales and resolutions. Moreover, InceptionV3 also includes a reduction block. These blocks use a combination of convolutional and pooling operations as well. Finally, a fully connected layer is established which maps out the output according to the desired number of classes to which a final softmax function is implemented to produce a class probability. The inception modules involved in InceptionV3 enables the network to capture features at different scales, enhancing the ability to recognize complex patterns and features. Moreover the use of this module also reduces the number of parameters making the model more efficient and functioning.

Advantages of InceptionV3

Inception V3 is good at understanding different details in pictures at the same time by using its building blocks. This model also plays a competitive role on benchmark datasets like Imagnet and image classification tasks. Like other models, it has the transfer learning capability means by using small datasets this model can be pre-trained for a large dataset to perform for specific tasks.

4.2 Hardware Configuration

Due to the size of our dataset along with the usage of heavy models, hardware implementation was necessary. We used AMD Ryzen 9 5950X 16-Core Processor, Nvidia RTX 3080 Ti graphics card along with 64GB RAM. According to our graphics processing unit (GPU), we installed the compatible CUDA version 11.2 and cuDNN Version 8.1. RTX 3080 Ti comes with a substantial amount of VRAM which is 12GB. It also significantly reduces training times and improves overall performance compared to less powerful GPUs and CPUs. This was proven when we initially tried to run our models through the GTX 1050 Ti graphics processing unit and Ryzen 3600 6-Core processor, a greater time was required which hampered the models performance as well. We used a deep learning framework such as Tensorflow which is optimized for GPU acceleration.

Chapter 5

Description of the dataset

5.1 Custom Data Collection Process

For this paper to be unique and specific, we focused on collecting our own dataset rather than using an existing dataset. As this paper is about accurately detecting a person at a random angle, we had to make sure that the range of each data point was restrictive and defined. Therefore, within the range of 0 to 180 degrees, we made 18 sectors with a 10-degree deviation. These angles were precisely divided by using a protractor, ruler and set square. In order to ensure the accuracy of these angles, a laser was also used. To capture the images, a Canon 750D DSLR camera was used.

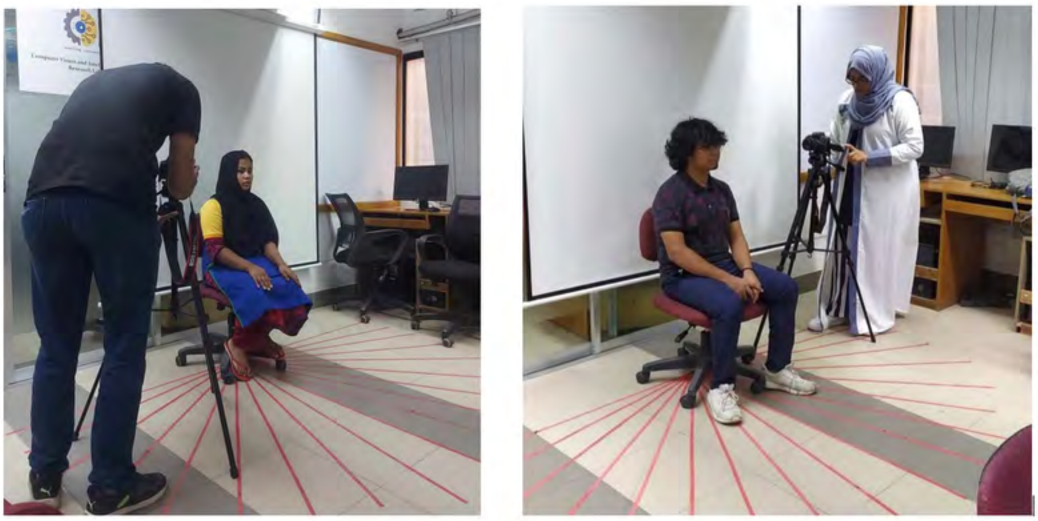


Figure 5.1: Data collection Setup

In this way, a total of 334 individuals images were collected. We were able to collect 6346 images of 334 individuals from different angles. To create a test dataset, an extra 19 images from 5 degree deviation was collected for 30 individuals. Hence, the total number of images for test dataset was 570. Figure 5.2 shows an example of one of the individual's data from the dataset and figure 5.3 shows the example of images from the test dataset.

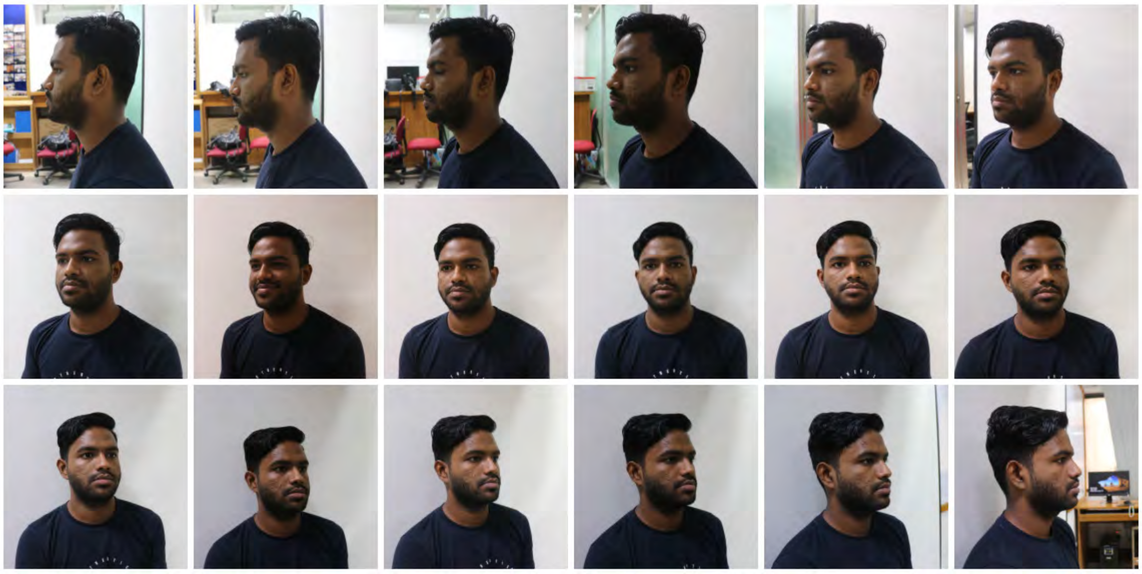


Figure 5.2: Raw data of an individual with 10 degree deviation

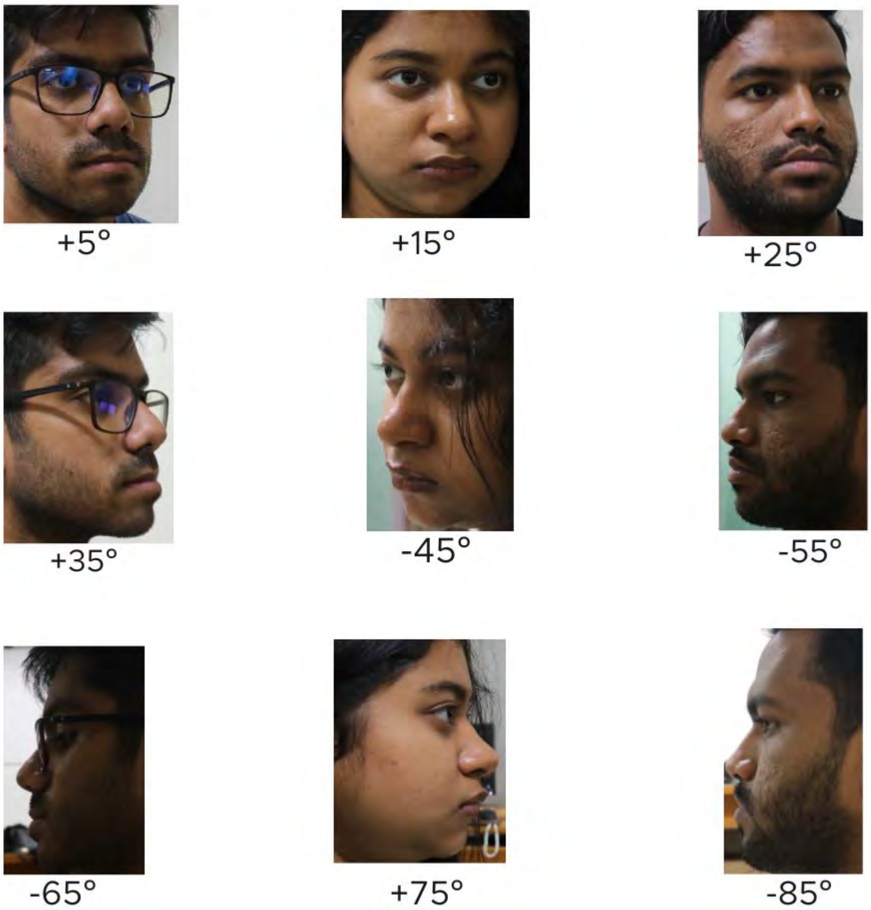


Figure 5.3: Test dataset images with 5 degree deviation

5.2 Pre-processing of data

5.2.1 Sorting and labelling of dataset

For classification of the entire dataset, folders for individuals had been created and named accordingly. As the dataset comprises 334 individual subjects, the folders had been named from ’1’ till ’334’. This folder contains images from 18 different angles at a 10 degree deviation and one from the centre. These 19 images had to be labelled according to the angles at which they were captured. We have initialized the center picture from 0 degree deviation as it is parallel to the face. Then, the right of the subject was considered to be positive and the left side to be negative. Therefore, each picture was labeled at its deviated angle along the 0 to 90 degree mark, for example “137_-30” or “137_+40” and so on. The file naming process couldn’t be automated because the images were angular and had to be named accordingly.

5.2.2 Face detection and facial landmark using MTCNN algorithm

- Face Detection: MTCNN identifies the faces in an image through a CNN at first stage. It gives the coordinates of bounding boxes that contain the detected faces.

- Facial Landmark Detection: Once the faces are detected, MTCNN takes it to the second stage where different facial landmarks within each of these face regions are identified. Typically, these landmarks are the positions of eyes, nose and mouth.

- Bounding Box Refinement: In the last stage, the algorithm fine tunes bounding boxes around detected faces in order to make sure of precise localization.

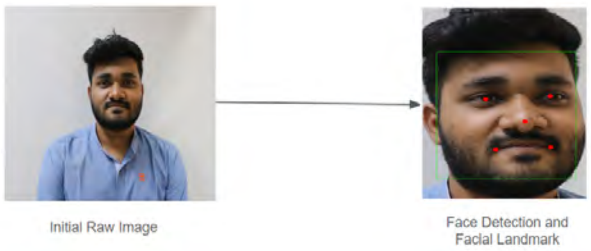


Figure 5.4: Face detection and facial landmark using MTCNN

5.2.3 Model input pre-processing

Resizing the images to one common size such as 224x224 pixels is necessary in order to pass them into fixed-size input models that we selected which include VGG19, VGG16, Resnet50, MobileNetv2 and DenseNet169. For Inceptionv3 model the input image had to be converted to 299x299 pixels. Data preprocessing steps such as normalizing and scaling pixel values ensure that the data will be handled efficiently by using a neural network. These methods enhance model performance and effectiveness in image classification tasks through consistency and efficient pattern recognition.

5.2.4 Data Augmentation

We used data augmentation using Keras ImageDataGenerator. Data augmentation is a vital method to improve model generalization while training neural networks for computer vision problems. Several augmentations are set up in the ImageDataGenerator, such as rescaling pixel values to the range [0, 1], adding random rotation within a given range, performing shear transformations, random zooming and random shifts and flips of the horizontal and vertical axes. These augmentations artificially broaden the training dataset, exposing the model to a greater variety of visual variances. The model's overall performance on unseen data is improved by adding these changes throughout the training phase, which makes the model more resilient and able to handle various object orientations, sizes and positions in real-world circumstances.

5.3 Data Classification

The dataset has been divided into 3 segments namely, training, validation and test.

- Training Set: A training set is a collection of data to train a model by providing relevant examples to learn about the types and patterns in order to make the ability to predict on new, unseen data.
- Testing Set: A testing set is another subset of data which is being used to evaluate the performance and prediction on completely new examples that wasn't used to train the model on training phase.
- Validation Set: A validation set is a separate subset of data that is used to evaluate and optimize a machine learning model's performance during training.

Number of images			
	Training set	Validation set	Testing set
Complete dataset (180°)	4442	1904	570

Table 5.1: Data Classification

We acquired a total of 6916 images. The percentage of training, validation and testing are 64.22%, 27.53% and 8.24% respectively.

5.4 Filtered Test Dataset

In order to analyse and study further, we filtered the test dataset and kept only the images captured between -60 to +60 deviation took along the 120 degree sector. Besides the complete test dataset, this has been created and named as "filtered test dataset". This reduces the complexity and simplifies the task and would make work easier for the model to identify as feature extraction becomes easier with images within 120 degree sector. By setting this to a restricted range, the model will most likely generalize better for variations within that particular angle. This approach most likely will bring about improved results than that of complete test dataset where images with more than +60/-60 degree deviation exists with minimal facial feature.

Chapter 6

Results & Analysis

6.1 Result Analysis

Different existing models for facial recognition were studied and used to train the data along with some customisation required for our dataset. Six models were trained and their respective accuracy Vs epoch graphs were generated along with validation accuracy, test accuracy, precision, recall and F1 score. All these were written in a table format below the section result comparison. And finally bar graphs were generated to show the improvements between the results of complete and filtered test dataset. Besides, the individual performance analysis of the respective models is also discussed below.

6.1.1 Confusion Matrix

The confusion matrix is a table that helps to evaluate the performance of classification algorithm on another set in which true values are known. It is especially effective in machine learning and statistics. The confusion matrix gives a clear and detailed breakdown of the model predictions and the actual outcomes, allowing a deeper analysis of performance. In most cases, a confusion matrix is represented as a square table where rows and columns are respectively to the actual class or category versus the predicted one. It provides a more detailed evaluation of the model. Moreover using the confusion matrix recall, F1 score and precision can be calculated. The four terms TP, TN, FP and FN are explained below:

True Positive(TP): are the correct predictions, the predictions which match with the real value.
True Negative(TN): these are negative observations which came out to be positive but are indeed negative.
False Positive(FP):these are predictions and observations which are actual results, but were confirmed as false.
False Negative(FN): these are results projected as negative although it is positive.

Accuracy: It is the proportion of instances which has been properly categorized over the number of all data instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

Precision: It is the ratio of true positives to the sum of true positives and false positives, it measures the accuracy of the positive predictions made by the models. This ensures the ability to avoid false positives. Whenever the result is positive, the result is likely to be correct.

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

Recall: It is the true positive or the sensitivity. Recall ratio of true positive by the summation of true positive and false negative. It assesses the models ability to capture all the positive instances and minimizes the false negatives. When the recall value is high, the model is effective in identifying the most positive case.

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

F1 score : it provides a balance measure that considers both false positives and false negatives. It is particularly important when there is an uneven class distribution in the dataset. The F1 score is defined as the harmonic mean of precision and recall, and is given by the formula:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6.4}$$

where *precision* is the number of correct positive results divided by the number of all positive results, and *recall* is the number of correct positive results divided by the number of positive results that should have been returned.

6.1.2 Performance Analysis of VGG-19 model

The below figure 6.1 shows the results of VGG-19 model. In this model, a batch size of 32 had been used which was run for 200 epochs. We initialized a learning rate of 0.0001. Data augmentation was also done with ImagedataGenerator. In order to customize the model, two dense layers had been used with 1024 and 512 neurons respectively. Besides, a ReLU activation function was implemented which helped with vanishing gradient problems. A single dropout of 0.7 has been initialized; this helped with regulation and prevented overfitting by reducing reliance on specific neurons. Batch Normalization is added to normalize the activation of the previous layer which helps in accelerating training and stability. Validation accuracy is monitored and only the best model observed during training based on the validation accuracy is saved. Moreover, after every 5 epochs, if there is no improvement in the validation loss, a factor of 0.2 is multiplied with the learning rate. A minimum learning rate threshold is set to 0.00001, this makes sure the learning rate does not reduce further. Moreover, to implement early stopping, the model analyzes the validation loss for 10 epochs if there is no improvement the training is halted. Lastly after fine tuning and modification the model successfully generated a test accuracy of 0.886 and a validation accuracy 0.7969 in 152 epochs.

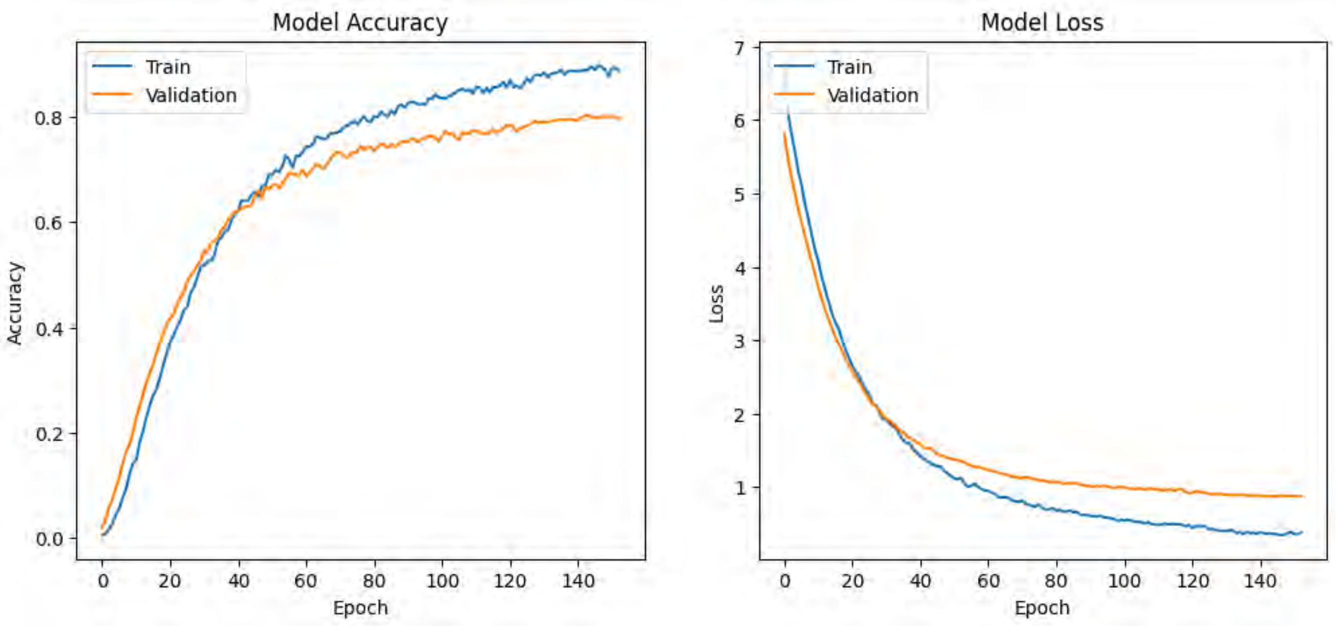


Figure 6.1: Accuracy and loss curves for VGG19

6.1.3 Performance Analysis of Resnet50 model

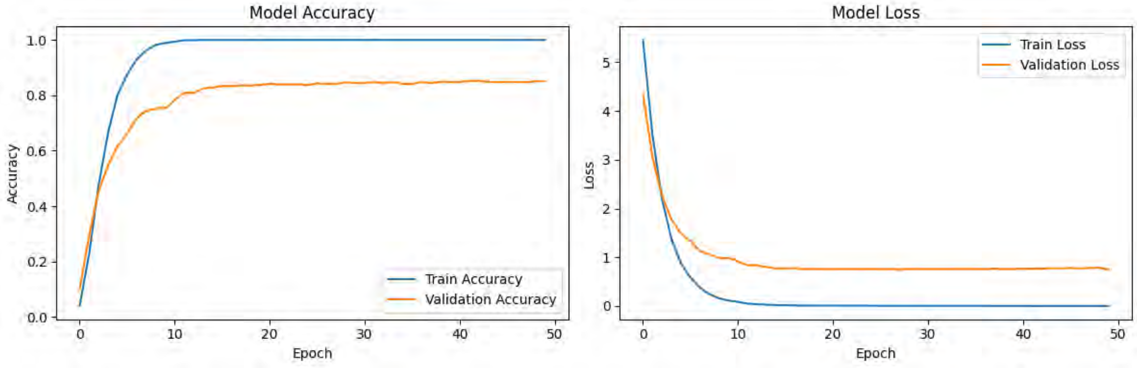


Figure 6.2: Accuracy and loss curves for Resnet50 model

The above figure 6.2 shows accuracy and loss curves of Resnet50 model's training and validation phase. A batch size of 32 with a learning rate of 0.0001 had been used which was run for 50 epochs. We added Dense and GlobalAveragePooling2D layers to the original model in order to modify the pretrained model. Initially, the fully connected layers at the top of the ResNet50 model were not included so that we can add our own custom layers. For layer customization we added a GlobalAveragePooling layer which reduces the spatial dimensions of the input to a single value per feature map. Then a dense layer of 1024 neurons along with ReLU activation function was added. Subsequently a final dense layer with the number of neurons equal to the number We achieved a validation accuracy of 0.81 and test accuracy of 0.8387. Lastly, we used Keras Checkpoint to save the model.

6.1.4 Performance Analysis of VGG16 model

Here, the figure 6.3 shows the results of the model. A batch size of 32 was used where the model is made to run for 200 epochs. We initialized the model with a learning rate of 0.0001. Data augmentation was also done with ImagedataGenerator. To customize the pretrained model, we added 2 dense layers, the first one consisting of 1024 neurons while the second consists of 512 neurons. A ReLU activation function also had been implemented. A dropout of 0.5 has been initialized, which allows random neuron selection to be ignored during training, this prevents overfitting. Batch Normalization is added to normalize the activation of the previous layer which helps in accelerating training and stability. Then another layer of dropout has been implemented. To personalize this model, we wanted to implement early stopping and reduce the learning rate. In this part, ReduceLROnPlateau which is a callback in Keras that allows the reduction of learning rate once the metric has stopped improving, this permits the model to converge faster and avoid overshooting.

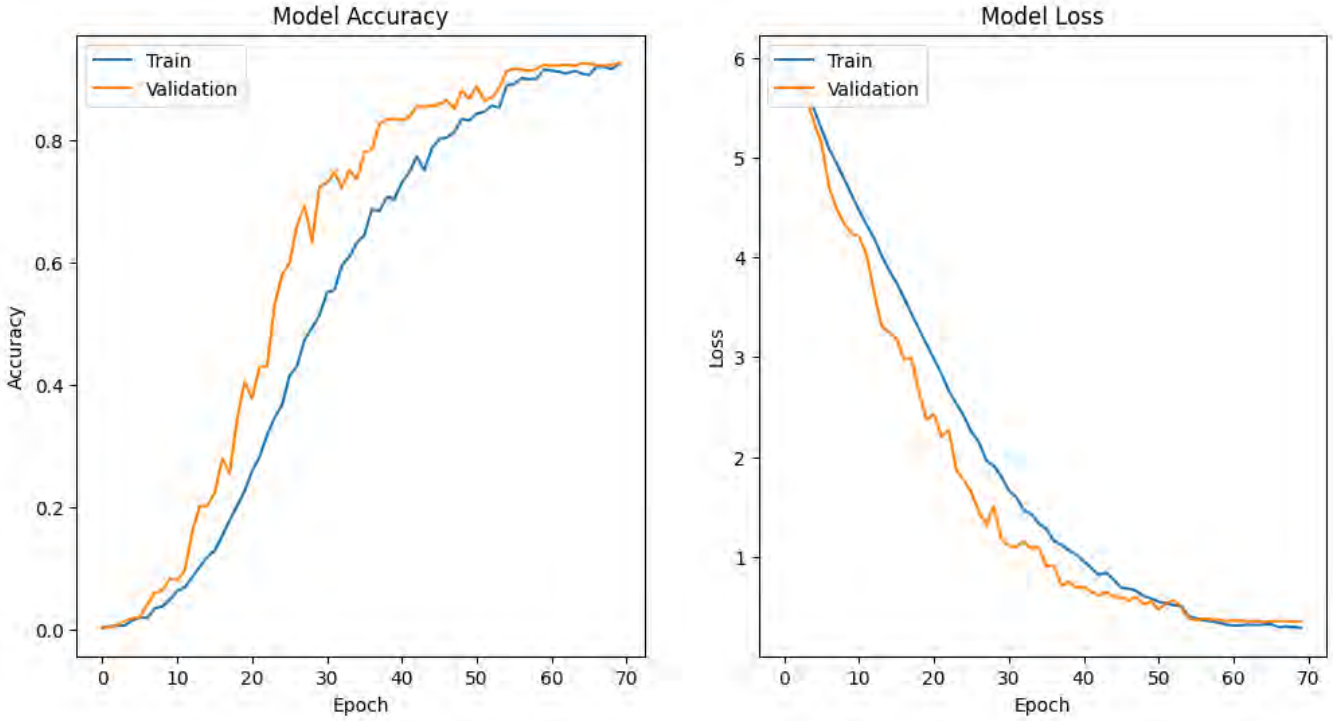


Figure 6.3: Accuracy and loss curves for VGG16 model

A variable named val_loss was assigned to monitor the validation loss. Moreover, after every 3 epochs, if there is no improvement in the validation loss, a factor of 0.2 is multiplied with the learning rate. A minimum learning rate threshold is set to 0.000001 this makes sure the learning rate does not reduce further. Besides, to implement early stopping, the model analyzes the validation loss for 5 epochs if there is no improvement the training is halted. Lastly due to fine tuning and customization, the model optimized itself and ran for 70 epochs and finished compiling yielding a test accuracy of 0.8622 and a validation accuracy of 0.9259.

6.1.5 Performance Analysis of MobileNetv2 model

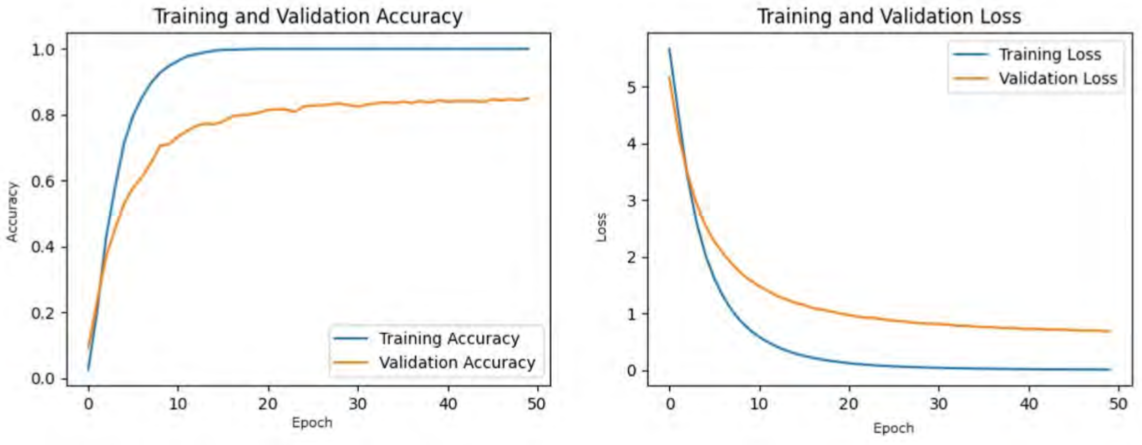


Figure 6.4: Accuracy and loss curves for MobileNetv2 model

The above figure 6.4 shows the accuracy and loss curves of the MobileNetv2 model's training and validation phases. A batch size of 32 with a learning rate of 0.0001 had been used which was run for 50 epochs. To change the pretrained model, we incorporated Dense and GlobalAveragePooling2D layers into the original model. In order to allow us to add our own custom layers, the fully connected layers at the top of the MobileNetv2 model were initially excluded. We provided a GlobalAveragePooling layer to configure the layers, which sets a limit of one value per feature map for the spatial dimensions of the input. Next, a dense layer of 1024 neurons was created, along with the ReLU activation function. Next, with as many neurons as there were classifications in our dataset, a final dense layer was constructed. We achieved a validation accuracy of 0.8497 and a test accuracy of 0.8480. Finally, we used Keras Checkpoint to save the model.

6.1.6 Performance Analysis of InceptionV3 model

The figure 6.5 shows accuracy and loss curves of InceptionV3 model's training phase. To make the InceptionV3 model more customizable, it is first supplied with pretrained weights from the ImageNet dataset but without its top layers. This model is then expanded with new layers designed particularly for face recognition: a global average pooling layer to minimize dimensionality and prevent overfitting, followed by a dense layer with 1024 units and ReLU activation to learn higher-level features. The final layer is a dense layer with a softmax activation function, with the number of units proportional to the number of classes (or persons) to be identified. This newly constructed model, which combines the original InceptionV3 model with the extra layers, is configured so that the base model's layers are "frozen". It means that these layers will not change during training, allowing the model to preserve previously learnt ImageNet features while training solely on the newly added layers. For the training phase, a learning rate of 0.0001 is set and employed in an RMSprop optimizer, which is an adaptive learning rate approach that aids in model tuning.

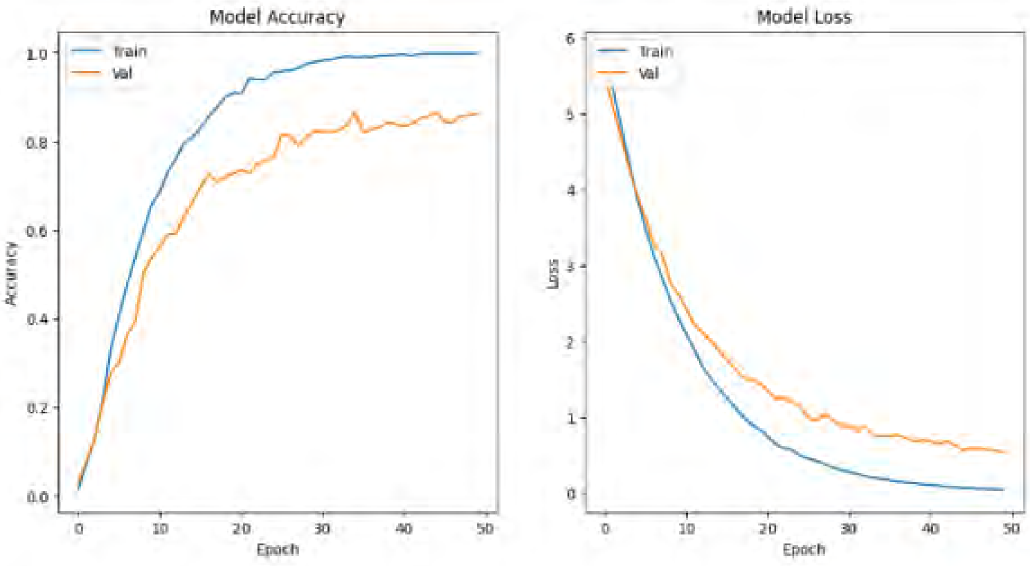


Figure 6.5: Accuracy and loss curves for InceptionV3 model

Finally, this optimizer compiles the model, with categorical cross entropy as the loss function (ideal for multi-class classification) and accuracy as the performance parameter. Finally the model is made to run for 50 epochs with a batch size of 32. A validation accuracy of 0.8616 and a test accuracy of 0.8413 was acquired.

6.1.7 Performance Analysis of DenseNet169 model

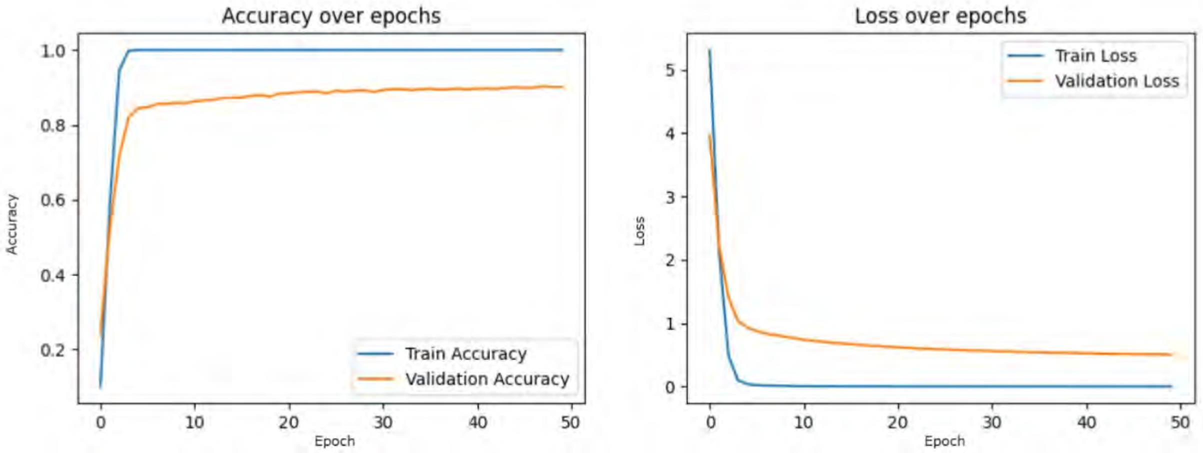


Figure 6.6: Accuracy and loss curves for DenseNet169 model

The above figure 6.6 shows accuracy and loss curves of DenseNet169 model's training phase. The DenseNet169 model is first built using pre-trained ImageNet weights, with the exception of the top layer, which is utilized to support a custom output layer for a given classification. The model has an input image size of 224x224 pixels in three channels (RGB). The DenseNet169 output is then flattened into a vector and fed through a dense layer of 1024 neurons using the ReLU activation function. This is followed by another dense layer, whose size is proportional to the number of classes in the training dataset, and which employs the softmax activation function for multi-class classification. The base DenseNet169 model's layers are frozen, so their weights will not change throughout training. This allows the model to keep its pre-trained features while only improving the weights of newly added dense layers. The model is then made with Adam optimizer with a learning rate of 0.0001 and a batch size of 32. The model was made to run for 50 epochs and a test accuracy of 0.8531 and a validation accuracy of 0.9012 was achieved.

6.2 Result Comparison

The table 6.1 presents a comparative evaluation of six deep learning models on a complete dataset, focusing on key performance metrics. VGG19 emerges as the top performer when it comes to test accuracy. It is most effective at correctly classifying test data. However, Resnet-50 leads in recall, it is able to correctly identify most positive instances. This is crucial in scenarios where missing a positive is costly. VGG19 shows the highest precision. This makes VGG19 the model of choice when false positives bear a higher cost. The F1 score, which balances precision and recall, is topped by VGG-19, Resnet-50, and MobileNetv2, all offering scores around 0.87. This indicates the robustness in handling both false positives and negatives.

Comparison Table with complete dataset				
Models	Test Accuracy	Recall	Precision	F1 score
VGG-19	0.886	0.867	0.877	0.87
VGG-16	0.862	0.843	0.851	0.85
Resnet-50	0.838	0.878	0.862	0.87
MobileNetv2	0.848	0.848	0.876	0.86
Inceptionv3	0.841	0.824	0.833	0.83
DenseNet169	0.853	0.841	0.860	0.85

Table 6.1: Result comparison between all the models using complete dataset

Comparison Table with filtered dataset				
Models	Test Accuracy	Recall	Precision	F1 score
VGG-19	0.992	0.940	0.950	0.95
VGG-16	0.941	0.924	0.916	0.92
Resnet-50	0.972	0.970	0.970	0.97
MobileNetv2	0.878	0.881	0.906	0.87
Inceptionv3	0.882	0.864	0.876	0.89
DenseNet169	0.941	0.920	0.933	0.93

Table 6.2: Result comparison between all the models using filtered dataset

The performance of different machine learning models on both full and filtered datasets is compared in the two tables that are provided. The filtered dataset results are shown in table 6.2 which indicates an overall improvement as well and makes it apparent that all the models show better performance metrics when compared between filtered dataset. For instance, VGG19 accuracy is boosted from 0.886 on the full dataset to an exceptional value of 0.992 for a filtered one, this instance is the same for the F1 score as it is also increasing in similar fashion. Resnet50 also demonstrates noticeable improvements, registering 0.972 in accuracy as well recall and precision on the filtered set of data. This performance trend is displayed in all models. This indicates that the filtration process have eliminated noise and irrelevant data. As a result, better learning and generalization was possible with the images of filtered dataset of 120 degree deviation.

6.2.1 Graphical representation of test accuracy and F1 Score

The graph in figure 6.7 compares the test accuracy of the 6 models at 180 and 120 degree sectors. VGG19 shows the highest accuracy in the 120 degree sector at 0.992 whereas MobileNetv2 has the lowest accuracy at only 0.882. On the other hand, in the 180 sector, DenseNet169 shows the highest test accuracy at 0.893 and Resnet50 has the lowest test accuracy at only 0.838.

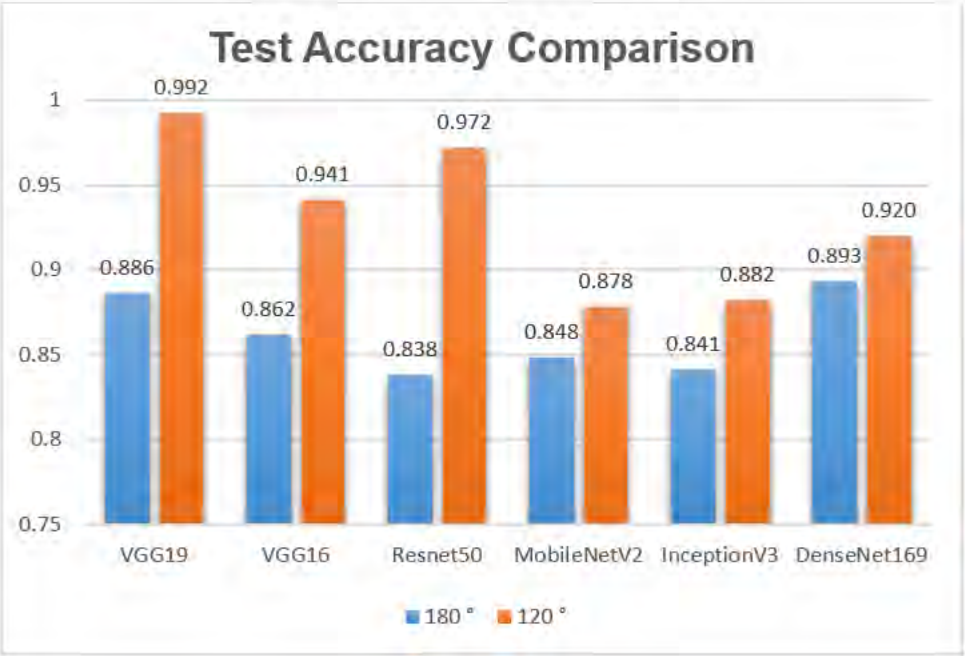


Figure 6.7: Test accuracy comparison

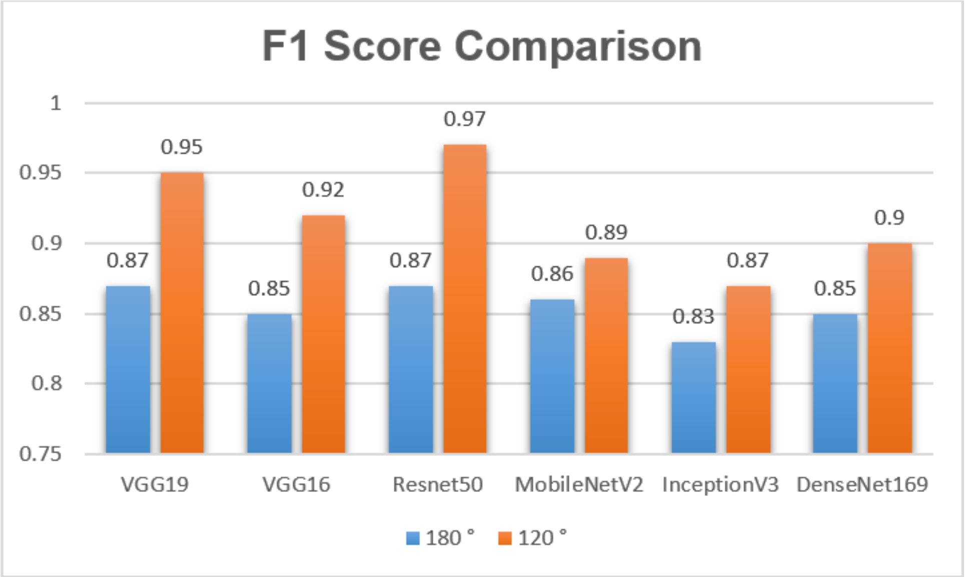
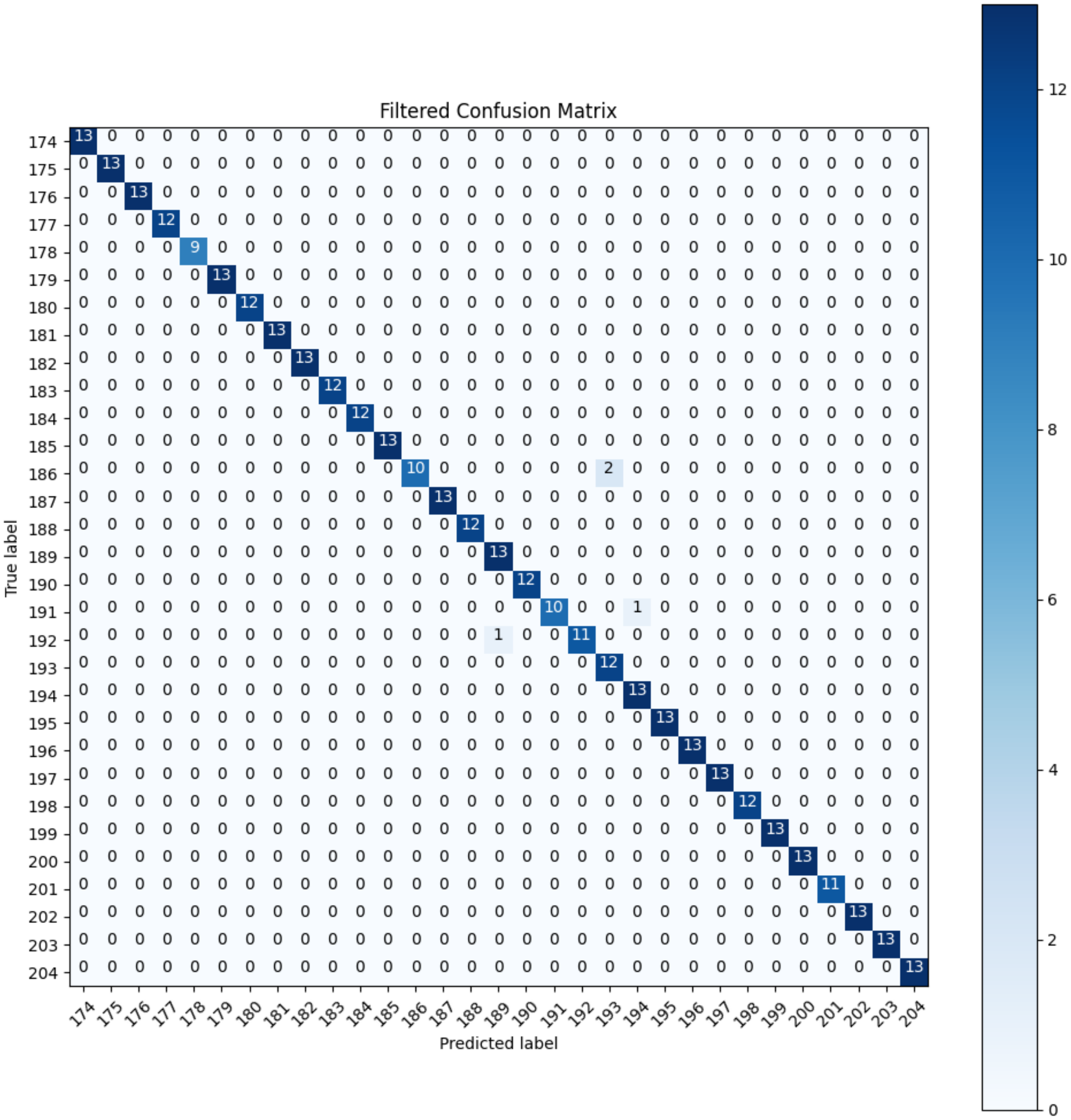


Figure 6.8: F1 score comparison

F1 scores of the models were also compared and shown in the figure 6.8. Similar to the test accuracy, this also shows improvement for the filtered test dataset.

6.2.2 Confusion Matrix of most improved models

As Resnet50 and VGG19 model showed maximum improvement when tested with the filtered dataset, the confusion matrix as shown in figure 6.10 and figure 6.9 for the 30 classes of Resnet50 and VGG19 was generated.



	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204
174	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
175	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
176	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
177	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
178	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
179	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
181	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
182	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
183	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
184	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
185	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
186	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
187	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
188	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
189	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
190	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
191	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	1	0	0	0	0	0	0	0	0	0	0
192	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0
193	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0
194	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0
195	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0
196	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0
197	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0
198	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0
199	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0
203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0
204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13

Figure 6.9: Confusion matrix for Resnet50

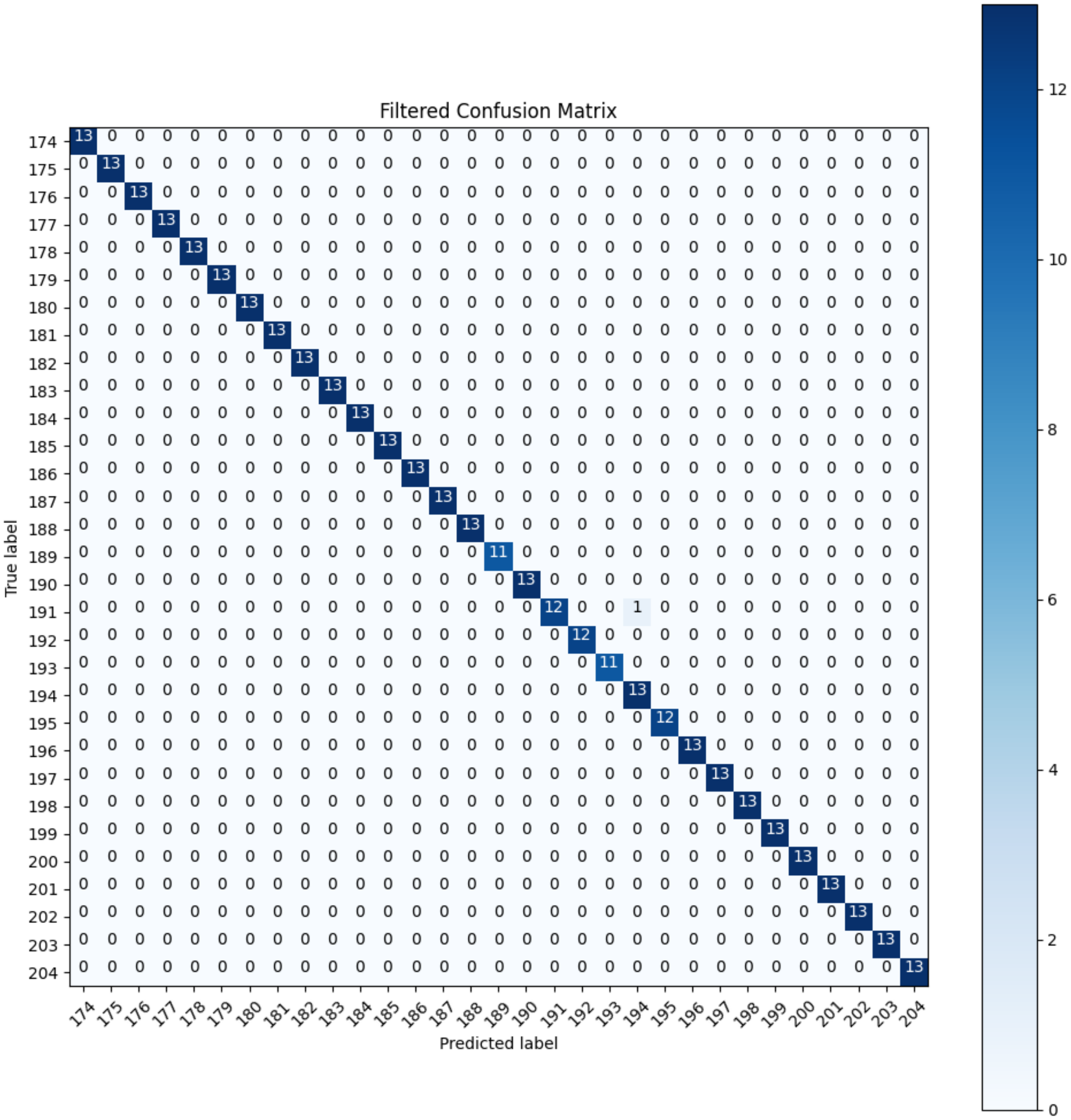


Figure 6.10: Confusion matrix for VGG19

6.3 Real-time and TensorRT

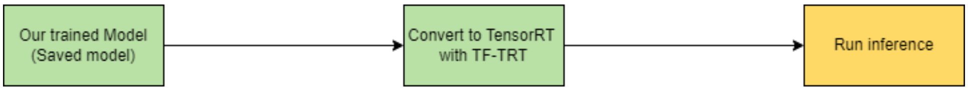


Figure 6.11: Converting our model to tensorRT

TensorRT is an effective technique that can greatly improve the effectiveness of facial recognition systems in the field of deep learning. TensorRT uses the power from hardwares including GPUs and specialized AI accelerators. To optimize and speed up the inference process which significantly reduces inference time. This is especially important for real-time applications where quick and precise identification are critical, such as facial recognition. This is made possible by optimization of the neural network model, layer fusion for increased parallelism, and use of calibration techniques to reduce calculations. Face recognition systems can therefore quickly process and analyze picture or video. This ensures quick and responsive detection even in high-throughput settings. We intend to convert our model to TensorRT as shown in 6.11 for better inference time.

Chapter 7

Future work & Conclusion

7.1 Future Work

In the future, our goal in successfully finishing our work is to implement our work in real time. This will take us to our main objective. We also intend to switch our model into TensorRT because it reduces inference latency considerably. This is important for real-time applications where low latency matters. Its optimization features, quantization options and runtime effectiveness make it the best option for implementing models particularly in resource limited settings where fast delivery and use of memory is a major concern. Moreover, we also plan to improve our models' performance and intend to develop an application integrating the best performing model in the future which will increase the responsiveness of our implementation in real time. This will allow our idea to be implemented and used in real life scenarios where facial recognition is necessary , like, in office or workplace, school and university and garments. In all these above scenarios, real-time detection at angular deviation proves to be of high regards.

7.2 Conclusion

Greater portion of the world population has been facing crime at public places and it is hard to determine the criminal hence walking towards justice might seem to be a prolonged path. Today's system is very slow and primitive hence our vision not only will help with identifying individuals in criminal settings which might be in movement, in a crowded setting but also help with keeping records of individuals in other crowded places such as offices, universities and school. Various techniques have helped in achieving an optimal output where facial recognition has been implemented. different size, shape and misalignment of the face are some common difficulties that arise during face recognition. Hence different systems could help in resolving them. Through our work, we have run some models in walking towards our goal of successfully detecting a subject using facial recognition techniques and compared the generated outcome. The models that had been implemented were, ResNet50, VGG16, VGG19, DenseNet169, MobileNetV2 and InceptionV3. Then a thorough comparison of the model has been done in order to utilize an optimal result.

Bibliography

[1] H. Zhang and Y. Guo, "Facial expression recognition using continuous dynamic programming," in *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, IEEE, 2001, pp. 163–167.

[2] B. Heisele, P. Ho, J. Wu, and T. Poggio, "Face recognition: Component-based versus global approaches," *Computer vision and image understanding*, vol. 91, no. 1-2, pp. 6–21, 2003.

[3] F. Wallhoff, M. Zobl, G. Rigoll, and I. Potucek, "Face tracking in meeting room scenarios using omnidirectional views," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, IEEE, vol. 4, 2004, pp. 933–936.

[4] K. Anderson and P. W. McOwan, "A real-time automated system for the recognition of human facial expressions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 1, pp. 96–105, 2006.

[5] M. Pantic and I. Patras, "Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 2, pp. 433–449, 2006.

[6] C. Geng and X. Jiang, "Face recognition using sift features," in *2009 16th IEEE international conference on image processing (ICIP)*, IEEE, 2009, pp. 3313–3316.

[7] F. W. Wheeler, R. L. Weiss, and P. H. Tu, "Face recognition at a distance system for surveillance applications," in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, IEEE, 2010, pp. 1–8.

[8] J.-L. Chen, T.-S. Hwang, and P.-W. Chang, "Cloud dynamic target-tracking and real-time 3d face-recognition by using bionic-vision capability," in *2012 International Conference on System Science and Engineering (ICSSE)*, IEEE, 2012, pp. 167–172.

[9] S. Berretti, A. Del Bimbo, and P. Pala, "Automatic facial expression recognition in real-time from dynamic sequences of 3d face scans," *The Visual Computer*, vol. 29, pp. 1333–1350, 2013.

[10] R. Naik and K. Lad, "Face recognition from multi angled images," *International Journal Of Engineering Research & Technology (Ijert) Issn*, pp. 2278–0181, 2015.

[11] J. I. Olszewska, "Automated face recognition: Challenges and solutions," *Pattern Recognition-Analysis and Applications*, pp. 59–79, 2016.

[12] M. S. I. Sameem, T. Qasim, and K. Bakhat, "Real time recognition of human faces," in *2016 International Conference on Open Source Systems & Technologies (ICOSST)*, IEEE, 2016, pp. 62–65.

[13] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, IEEE, 2018, pp. 169–175.

[14] A. Sajjanhar, Z. Wu, and Q. Wen, "Deep learning models for facial expression recognition," in *2018 digital image computing: Techniques and applications (dicta)*, IEEE, 2018, pp. 1–6.

[15] X. Tian, H. Daigle, and H. Jiang, "Feature detection for digital images using machine learning algorithms and image processing," in *SPE/AAPG/SEG Unconventional Resources Technology Conference*, URTEC, 2018, D023S034R004.

[16] Y. Zheng, C. Yang, and A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography," in *Computational Imaging III*, SPIE, vol. 10669, 2018, p. 1 066 905.

[17] H. Chen and C. Haoyu, "Face recognition algorithm based on vgg network model and svm," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1229, 2019, p. 012 015.

[18] A. B. Perdana and A. Prahara, "Face recognition using light-convolutional neural networks based on modified vgg16 model," in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, IEEE, 2019, pp. 1–4.

[19] B. Police. "Crime statistics 2019 report." (2019), [Online]. Available: https://www.police.gov.bd/en/crime_statistic/year/2019.

[20] H. Ku and W. Dong, "Face recognition based on mtcnn and convolutional neural network," *Frontiers in Signal Processing*, vol. 4, no. 1, pp. 37–42, 2020.

[21] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, "Real-time and accurate drone detection in a video with a static background," *Sensors*, vol. 20, no. 14, p. 3856, 2020.

[22] S. Abbasi, H. Abdi, and A. Ahmadi, "A face-mask detection approach based on yolo applied for a new collected dataset," in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, IEEE, 2021, pp. 1–6.

[23] L. Ali, F. Alnajjar, H. A. Jassmi, M. Gocho, W. Khan, and M. A. Serhani, "Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures," *Sensors*, vol. 21, no. 5, p. 1688, 2021.

[24] H. Aung, A. V. Bobkov, and N. L. Tun, "Face detection in real time live video using yolo algorithm based on vgg16 convolutional neural network," in *2021 International conference on industrial engineering, applications and manufacturing (ICIEAM)*, IEEE, 2021, pp. 697–702.

[25] T. Gwyn, K. Roy, and M. Atay, "Face recognition using popular deep net architectures: A brief comparative study," *Future Internet*, vol. 13, no. 7, p. 164, 2021.

[26] G. Suguna, H. Kavitha, and S. Sunita, "Face recognition system for realtime applications using svm combined with facenet and mtcnn," *Int. J. Electr. Eng. Technol.(IJEET)*, vol. 12, pp. 328–335, 2021.

[27] M. Tamilselvi, S. Karthikeyan, and G. Ramkumar, "Face recognition based on spatio angular using visual geometric group-19 convolutional neural network," *Annals of the Romanian Society for Cell Biology*, pp. 2131–2138, 2021.

[28] K. Teoh, R. Ismail, S. Naziri, R. Hussin, M. Isa, and M. Basir, "Face recognition and identification using deep learning approach," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1755, 2021, p. 012 006.

[29] Z. Zhang, H. Zhang, H. Liu, S. Xin, N. Xiao, and L. Zhang, "Frontal face generation based multi-angle face identification system," in *2021 International Conference on Computer, Control and Robotics (ICCCR)*, IEEE, 2021, pp. 329–334.

[30] M. Chowdhury. "What is the importance of facial recognition in today's world?" (2022), [Online]. Available: https://www.analyticsinsight.net/what-is-the-importance-of-facial-recognition-in-todays-world/.

[31] C. Sikora. "Assaults, drug crimes on valley buses, light rail have risen in the last 5 years." (2022), [Online]. Available: https://www.12news.com/article/news/crime/assaults-drug-crimes-valley-buses-light-rail-risen-last-5-years/75-5de267bd-46f8-4ec9-a3c2-bed63ef74477.

[32] R. Gül. "How facial recognition is used in the world." (), [Online]. Available: https://www.cameralyze.co/blog/how-facial-recognition-is-used-in-the-world.

[33] S. L. M. Oo and A. N. Oo, "Child face recognition system with deep learning,"