Durban University of Technology

Development of a Face Mask Detection and Masked Facial Recognition Model based on a Hybrid Convolutional Neural Network

By

Chezlyn Pillay

21709836

A dissertation submitted in fulfilment of the requirement for the degree of Master's in Information and Communications Technology

Department of Information Technology,
Faculty of Accounting and Informatics

Supervisor: Dr Seena Joseph

Co-Supervisor: Professor Brett van Niekerk

2024

DECLARATION

I, *Chezlyn Pillay*, declare that:

(i) The research reported in this dissertation, except where otherwise indicated, is my original research.

(ii) This dissertation has not been submitted for any degree or examination at any other university.

(iii) This dissertation does not contain other persons' data, pictures, graphs, or other information, unless specifically acknowledged as being sourced from other persons.

(iv) This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

- Their words have been re-written, but the general information attributed to them has been referenced.
- Where their exact words have been used, their writing has been placed inside quotation marks, and referenced.

(v) This dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation and in the Reference Section of this dissertation.

Signature: Chezlyn Pillay **Date:** 29 January 2025

APPROVAL FOR FINAL SUBMISSION

Supervisor: Dr. Seena Joseph Signature: Date: 05/02/2025

Co-Supervisor: Prof. Brett van Niekerk Signature: Date: 03/03/2025

ACKNOWLEDGEMENTS

I am grateful to acknowledge the collective effort of several individuals that have played a pivotal role to the successful completion of the proposed study. With profound sincerity, I extend my gratitude to all those who have contributed to this endeavour. First and foremost, I attribute the completion of this research study to the grace of my faith in Jesus, which enabled me to navigate its complexities. My profound appreciation goes to my esteemed supervisor and co-supervisor, Dr. Seena Joseph, and Professor Brett van Niekerk. Their unwavering commitment, dedication, and substantial investment of time were pivotal in guiding me to produce a research paper of master's calibre. Dr. Joseph's insightful suggestions and invaluable counsel consistently illuminated the path forward, enriching this work significantly. The guidance both Dr Seena and Professor Brett provided was indispensable, and without their mentorship, the realisation of this dissertation would have remained beyond reach. Furthermore, my heartfelt appreciation goes to the members of my family, Mervyn, Tracy, Kerion, Schae, and Kathleen, as well as my circle of friends for their unwavering support. Their continuous encouragement, support, and motivational influence played a critical role in propelling me to create a dissertation of commendable quality. Their enduring love and assistance throughout this journey has served as a pillar of strength and motivation.

ABSTRACT

In the context of enhancing security authentication, facial recognition technology has become pivotal, replacing conventional authentication methods such as passwords, security tokens and PINs across various sectors. However, the rapid growth of facial recognition technology faced hindrances due to the COVID-19 pandemic, where mandatory face mask usage obscured facial features, challenging existing authentication methods. Regardless, the existence of several methods for face mask detection and recognition highlighted prevalent issues such as poor lighting, varied angles, failed detection for improper use of face masks, computational complexity, difficulty in detecting smaller faces and low-resolution targets have led to suboptimal accuracy rates. Hence, this study aims to address these challenges by introducing a hybrid Convolutional Neural Network (CNN) architecture tailored for Face Mask Detection (FMD) and Masked Facial Recognition (MFR). The models used MobileNetV2 and FaceNet InceptionResNetV1 respectively for FMD and MFR. The proposed models leverage advanced FMD and MFR technologies, contributing to the real-world need for enhanced security in scenarios where traditional methods are insufficient. The models underwent training using five distinct datasets, comprising a total of 10,980 images for FMD across two datasets, and 26,523 images for MFR across three datasets. In the FMD phase, the model achieved exceptional results, attaining a perfect 100% across evaluation metrics such as accuracy, precision, recall, and the f1-score within a training timeframe of an hour. Transitioning to the MFR phase, where the model required approximately one hour and 30 minutes, maintained an outcome of 99.68% across the aforementioned metrics, surpassing the accuracy level of existing models within the meta-analysis. Furthermore, the model underwent testing on a real-time custom dataset designed for MFR evaluation, consisting of 5500 images (i.e., 4400 for training, 550 for validation and 550 for testing) in real-life scenarios. Robustness was assessed under various conditions, resulting in an impressive 99.82% accuracy. The model demonstrated high accuracy in real-time testing. Notably, both the models excel in detecting and recognising masked participants from diverse angles and lighting conditions with minimal computational complexity. Leveraging the pre-trained MobileNetV2 for FMD and FaceNet InceptionResNetV1 with CNN for MFR, the CNN models provide a comprehensive solution. The proposed models surpass existing methods, excelling in accuracy under challenging conditions. This study contributes a versatile and efficient solution, addressing limitations in current approaches and providing robust models for FMD and MFR in diverse sectors.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter Two

Chapter Three

Chapter Four

Chapter Five

LIST OF TABLES

Chapter Three

Chapter Five

CHAPTER ONE: INTRODUCTION

1.1. Background

Technology has undergone a transformative shift due to substantial technological progress over the years. Amidst this transformative shift, facial recognition is gaining traction as a preferred method of authentication due to its convenience, efficiency, and increasing accuracy in diverse applications (Rahmani *et al.* 2022). These applications include but are not limited to features such as unlocking smartphones, accessing secure facilities, and authenticating payments (Qinjun *et al.* ; Seng, Al-Ameen and Wright 2021). Within the realm of facial recognition advancements, the automated processes of Face Mask Detection (FMD) and Masked Facial Recognition (MFR) have emerged prominently. These technologies employ machine or deep learning to discern whether individuals are wearing face masks or not, a development accentuated by the widespread impact of the COVID-19 pandemic (Hussain *et al.* 2022; Ullah *et al.* 2022). The ongoing usage of face masks, which initially surged in response to the COVID-19 pandemic, has now expanded to encompass preventive measures against various respiratory illnesses (Das, Ansari and Basak 2020; Desai and Mehrotra 2020; Dharanesh and Rattani 2021), as well as in various other scenarios. These scenarios include but are not limited to crime prevention, hospital settings, and pollution-prone environments (Wojcik and Austin 2020; Kodros *et al.* 2021).

Since traditional facial recognition determines the identity of an individual's face, conversely, wearing a face mask obscures the accuracy in determining the correct identity of an individual (Sharma 2020). Therefore, the adoption of face masks pose a concern in traditional facial recognition systems as it extensively conceals a substantial portion of the facial region, rendering crucial features such as the mouth, nose, and facial structure unobservable (Alzu'bi *et al.* 2021). Hence, the implementation of an effective and efficient FMD and MFR system is becoming an essential component for an accurate authentication mechanism for various sectors such as public health, education, banking and safety, and security (Wojcik and Austin 2020; Kodros *et al.* 2021). In addition, this model can aid in providing compliance where necessary whilst ensuring safety in certain environments where face mask protocols are compulsory.

The accurate and efficient detection of a face mask and the recognition of an individual wearing a face mask relies heavily on machine learning, deep learning and computer vision tools (Oumina, Makhfi and Hamdi 2020). Machine learning algorithms have been widely employed in this context. These include the Support Vector Machine (SVM) (Mundial et al. 2020), K-Nearest Neighbour (KNN) (Oumina, Makhfi, and Hamdi 2020), and Local Binary Pattern Histogram (LBPH) (Arya and Tiwari 2021; Mhadgut 2021; Suhaimin et al. 2021). Additionally, deep learning algorithms, such as CNN (Boulos 2021; Talahua et al. 2021; Ullah et al. 2022b), along with ensemble models (Mundial et al. 2020; Abbasi, Abdi, and Ahmadi 2021; Hariri 2022), have also been employed. These algorithms share the common objective of identifying critical facial features such as the eyes, jawline, forehead, and eyebrows. They are subsequently trained for accurate classification, enabling the determination of whether an individual is wearing a face mask and accurately identifying them despite the presence of the mask (Sharma 2020).

Within this context, the utilisation of Artificial Intelligence (AI) models, particularly the CNN, emerges as a cornerstone technology due to its ability to effectively analyse facial features despite mask occlusion (Christa *et al.* 2021). The CNN fundamentally represents a category of deep neural networks that consist of multiple layers, notably including an input layer, convolutional layer, pooling layer, fully connected layer and an output layer (Kaur *et al.* 2022). These networks leverage the backpropagation technique to grasp spatial patterns autonomously and efficiently within the provided data (Sun *et al.* 2017; Ullah *et al.* 2022; Naseri and Mehrdad 2023). The CNN is designed specifically to work with images and they are extremely parameter efficient since the kernels are shared across full image locations (Mohammed Ali and Al-Tamimi 2022). These characteristics have led to CNN becoming extensively accepted in computer vision for applications such as object detection and recognition (Naseri and Mehrdad 2023). User authentication-based applications frequently use facial recognition as a strategy for biometric authentication. Thus, CNN has demonstrated its effectiveness in facial recognition and classification (Islam *et al.* 2020; Mundial *et al.* 2020; Wang *et al.* 2020; Boulos 2021; Talahua *et al.* 2021; Hariri 2022; Ullah *et al.* 2022). The CNN models therefore, when tailored to specific face detection and recognition tasks, offer unparalleled accuracy and reliability in navigating the complexities of FMD and MFR scenarios, consequently leading to advancements in this critical domain (Islam *et al.* 2020).

1.2. Research Problem

The emergence of face masks pose challenges for traditional facial recognition systems, necessitating the development of masked facial recognition models. However, despite advancements in FMD and MFR models, significant challenges persist in this realm. Within the context of the current FMD and MFR models such as the LBPH (Arya and Tiwari 2021; Suhaimin *et al.* 2021), You Only Look Once (YOLOV) (Aswal *et al.* 2020; Mhadgut 2021), Visual Geometry Group (VGG) models (Wei *et al.* 2020; Hariri 2022), the Eigenface (Kadhim, Jabber and Hadi 2019), FaceNet (Cahyono, Wirawan and Rachmadi 2020), and Fisherface (Jayaswal and Dixit 2020; Reddy and Kumar 2021), challenges remain in accurately identifying masked faces, resulting in decreased precision and a higher rate of false positives (Fang et al., 2019; Ding et al., 2020). Efforts to develop an efficient algorithm for FMD and enabling MFR have therefore faced critical challenges. These challenges include the model's vulnerability to changes in lighting conditions, which render it ineffective (Islam *et al.* 2020; Arya and Tiwari 2021; Talahua *et al.* 2021), and the impact of changes in varied angles (Islam *et al.* 2020; Arya and Tiwari 2021; Talahua *et al.* 2021) of the human face, making the model incapable of recognising individuals. A majority of the developed models including but not limited to YOLOV, LBPH, VGG, AlexNet, ResNet-50 and MobileNetV2, primarily focus on FMD, emphasising the necessity of addressing the integration of MFR (Islam *et al.* 2020; Yang *et al.* 2020; Hariri 2022). A substantial number of developed models, encompassing various approaches such as YOLOV, Eigenface, LBPH, VGG, AlexNet, ResNet-50, SVM, and MobileNetV2 display an accuracy rate of 95% or below for FMD and MFR (Das, Ansari and Basak 2020; Damer *et al.* 2021; Ullah *et al.* 2022). Current FMD and MFR models, such as MobileNetV2, FaceNet, and YOLOV, struggle with the ability to detect smaller faces and low-resolution targets due to occlusions and noise (Hussain et al. 2021; Rafidison et al. 2023). Models, including LBPH, Multi-Layer-Perceptron (MLP), and CNN face similar challenges, exacerbated by occlusions and noise, affecting their performance in FMD and MFR (Talahua et al. 2021; Rahmani et al. 2022; Rafidison et al. 2023). Additionally, models such as CNN, MobileNetV2, MLP, YOLOV, and ResNet encounter difficulties in distinguishing between incorrect mask usage and no mask at all, failing to address improper mask usage (Cimmino et al. 2022; Sheikh and Zafar 2023). Therefore, the highlighted challenges encountered in developing effective models for FMD and MFR emphasise the urgency and need for the development of an innovative FMD and MFR solution.

1.3. Research Aim and Objectives

The aim of this study is to develop a FMD and MFR model based on CNN to detect and recognise masked faces. The aim can therefore be achieved by ensuring the following objectives are met within the research:

i. To critically analyse the existing literature using the Systematic Literature Review (SLR) based on the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) protocol to identify the current trends in FMD and MFR models.
ii. To develop a hybrid CNN based FMD and MFR model to classify and recognise masked faces.
iii. To evaluate the performance of the newly developed FMD and MFR models by comparing them against the existing models using well-known evaluation metrics.

1.4. Contribution of the Study

The research study conducted is aimed at producing a functioning model that is developed through the utilisation of the CNN algorithm. The model is tailored to detect face masks and conduct MFR, reinforcing security across multiple sectors, including companies, government organisations, academic institutes, hospitals, and the banking sector (Memon 2017; Alzu'bi et al. 2021). This addresses the aforementioned limitations of current FMD and MFR models.

The research endeavour seeks to adopt the development of a model that works effectively in the midst of adverse effects of change in lighting conditions that render the model ineffective. The research output also considers the varied angles of the human face, as this factor can hinder the model's recognition capabilities for individuals. This study aims to substantially enhance the accuracy rates of both FMD and MFR models. In addition, the proposed model intends to reduce computational complexity while addressing incorrect mask usage detection and effectively handling images with occlusions. Furthermore, as a significant contribution to the study, a SLR meta-analysis is conducted following the PRISMA protocol guidelines (Liberati *et al.* 2009), offering a comprehensive synthesis of existing research findings in the field of FMD and MFR. This meta-analysis serves to enrich the study's insights and provide a robust foundation for further analysis and discussion (Liang *et al.* 2020).

Ultimately, the research output aims to develop a versatile model, incorporating the SLR PRISMA protocol, and proficiently managing both FMD and MFR tasks. By providing a robust and precise solution, the development of this model holds the promise of reshaping how organisations approach both public health and security, marking a pivotal step towards a safer and technologically advanced future (Kheaksong et al. 2022).

1.5. Structure of the Dissertation

A coherent progression of concepts is followed in the organisation of this dissertation. Chapter one initiates with a comprehensive preface to the dissertation. It discusses and describes the research background, problem statement, its aims, and objectives, and the contribution of the proposed study.

Subsequently, chapter two meticulously outlines the theoretical framework utilised for the development of the respective models. This chapter introduces vital mathematical concepts and steps essential for developing the CNN models, highlighting key frameworks. It specifically explores variations, emphasising the significance of the CNN architecture and pre-trained models such as MobileNetV2 and FaceNet InceptionResNetV1 utilised in constructing the FMD and MFR models.

Within the third chapter, an extensive examination of the pertinent literature is presented. This section delves into an exploration based on the diverse array of FMD and MFR models that have emerged due to the influence and impact of the COVID-19 outbreak and other respiratory illnesses. Using the PRISMA protocol, a systematic review along with a detailed meta-analysis is conducted. This highlights the algorithms used to construct the models, their accuracy rates, metrics, trends in FMD and MFR approaches, methods, model types, number, and type of datasets utilised. Furthermore, chapter three provides a detailed description of the limitations identified within the existing literature studies.

In chapter four, the methodology employed for the proposed study in context of model development is discussed. This includes a comprehensive explanation regarding the methods and materials used for the proposed model architecture. In addition, this chapter aims to comprehensively discuss the process flow necessary to execute the development of both models.

An in-depth analysis and evaluation of both the developed models' performance are presented in chapter five. It further discusses and analyses the output of all results for the required experiments whilst comparing the experiment results with the existing literature results identified. The outcomes generated are examined to assess the models' performance, focusing on the relevant evaluation metric percentage scores of the selected models.

The dissertation concludes with chapter six, where a comprehensive summary of the study's contributions, observations, limitations, and prospective areas for enhancement are presented. This provides valuable information and enhances the collective knowledge base to lay groundwork for subsequent research endeavors.

CHAPTER TWO: THEORETICAL FOUNDATION OF THE CONVOLUTIONAL NEURAL NETWORK

This chapter is used to present the core frameworks essential to the development of the Convolutional Neural Network (CNN) model by introducing the key mathematical concepts and steps required to build a robust model. The chapter further describes the variations of the CNN framework with specific attention to both the MobileNetV2 and FaceNet InceptionResNetV1 model, imperative to the development of the Face Mask Detection (FMD) and Masked Facial Recognition (MFR) models. This chapter therefore provides a theoretical background based on the foundational building blocks of the models whilst unravelling the intricate patterns and processes of these core models.

2.1. Architecture of the Convolutional Neural Network Algorithm

Within the field of deep learning the CNN has emerged as a significant advancement in the realm of Artificial Intelligence (AI) (Naseri and Mehrdad 2023). Convolutional neural networks are designed to extract intricate features from data through hierarchical learning, similar to that of the visual cortex (Chen, Cui and Ding 2023). There are several layers in a CNN, including convolutional layers that extract features, pooling layers that reduce dimensionality, and fully connected layers that classify data. Through the utilisation of the CNN architecture, automatic pattern identification is produced and therefore excels in applications that require object detection, image segmentation and image classification (Kaur et al., 2022). Understanding the architecture and functioning of the CNN is of paramount importance in order to build an efficient FMD and MFR model within the field of computer vision. The choice of CNN adoption for FMD and MFR offers streamlined advantages, eliminating specific preprocessing requirements such as feature alignment (Kaur et al., 2022). With proven value in image processing, this algorithm excels in both FMD and MFR, minimising errors and ensuring higher accuracy (Khan et al., 2019). Additionally, CNN facilitates the use of pre-trained models, enabling transfer learning and expediting development (Sheikh et al., 2018; William et al., 2019; Yang, Ge, and Zhang, 2020; Wu and Zhang, 2021).

The convolutional neural network consists of the basic components that are required to work in a systematic manner. These components produce a functioning model that can provide efficient data

visualisation (Kiranyaz *et al.* 2021). The CNN consists of the basic convolutional layers which through convolution operations, extract features from the input images (Ayyadevara 2018). In these layers, spatial patterns such as edges and textures are captured by filters or kernels sliding across the input image. Once the convolutional layer processes have been completed, the pooling layers are then applied (Naseri and Mehrdad 2023). In the pooling layers, feature maps are down sampled in order to reduce the spatial dimension of an image, thus resulting in reduced computational complexity based on the input convolutional layers (Yang *et al.* 2019). Therefore, the pooling layer processes ensure only the most pertinent information is retained. Subsequently, the fully connected layers are responsible for classification, where features are extracted and fed into a dense neural network to make predictions (Islam *et al.* 2020; Naseri and Mehrdad 2023). While the input and output layers are essential components, these layers are fundamental to the CNN architecture and serve as the entry and exit points for data flow, respectively (Naseri and Mehrdad 2023). Through the process of each of these layers working together, the CNN can automatically learn intricate and distinct patterns enabling it to identify these patterns from a basic input image (Kiranyaz *et al.* 2021). Based on *Figure 2. 1*, a comprehensive explanation is presented for each layer of the CNN architecture. These layers consist of:

- Input Layer
- Convolution Layer (filter/kernel process)
- Pooling Layer
- Fully Connected Layer
- Output Layer

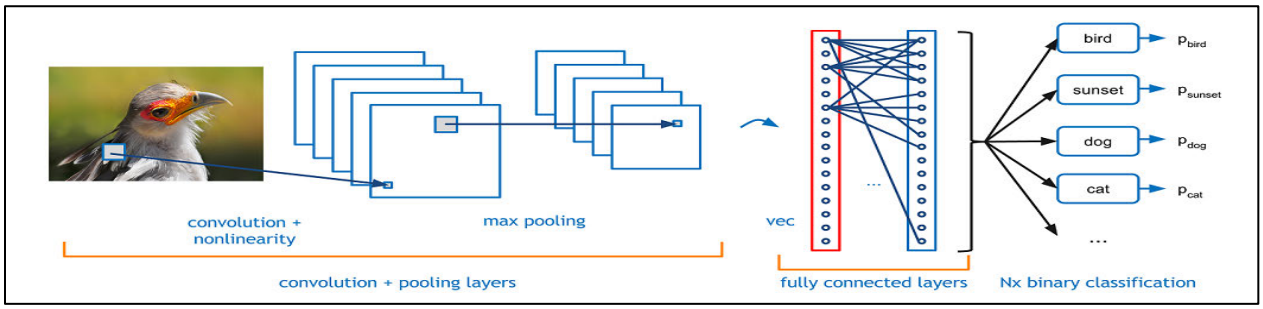


Figure 2. 1: A Demonstration of the CNN Architecture for Image Classification (Singh 2023b).

2.1.1. Input Layer

Initiating the process, involves the introduction of the basic input image. The operation commences when the basic input image is introduced to the CNN. The CNN begins by searching for specific features in order to learn and easily identify intricate patterns within the image. This process is conducted in order to identify all images within the specified dataset that possess the same attributes (Hao *et al.* 2020). Upon the identification of a specific feature in the input images, the first layer of filters "activate" (or compute high values) around the image (Naseri and Mehrdad 2023). The input image is then processed by multiple filters, mapped, and learned one by one as each filter slices through it (Sun *et al.* 2017). Since every layer searches for and inherently captures a pattern, for example, patterns such as corners, dots and edges are captured by the first layer of the filters, smaller patterns are then combined into larger ones in the subsequent layers (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023)

2.1.2. Convolutional Layer

In the process of the convolution layer following the input image layer, a matrix is applied to the input image whereby a mathematical operation is performed using integers to manipulate the images (Naseri and Mehrdad 2023). The computation process is calculated and resulted by combining all neighbouring weighted values together to identify the most central pixel (Naseri and Mehrdad 2023). Hence, a new filtered and modified image is formed as the new output. Using a matrix, a convolution operation is performed by multiplying a pixel's colour value with the colour values of its neighbours (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). Since a filter is employed to extract the most pertinent features of an image, it can be perceived and utilised as a matrix to traverse the data inputted into the process (Zebari *et al.* 2020). The process then commences with the conceptualisation of how neural networks function (i.e., known as a dot product) based on a specific sub region of the data that was input into the operation. Once the dot product has been performed, an output of the dot products form the matrix. The size of a kernel or filter is arbitrary in nature, despite the commonly utilised 3x3 matrix (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). In a Red, Green, and Blue (RGB) image the CNN accommodates for a three-dimensional (3D) image format. Hence, the computation performed is multiplied by three since it utilises three different pixel matrices (i.e., a red, green, and blue) (Ma and Yuan 2019). The height, width and channels are therefore multiplied

(Zebari *et al.* 2020). In a grayscale image format, the CNN operates on a two-dimensional (2D) image format (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). Thus, the computation is singular as it involves a single pixel matrix, representing various shades of grey (Cascianelli *et al.* 2018; Ma and Yuan 2019). Within the grey scale format only the width and height of the image are considered, without the inclusion of additional channels required for the pertinent colour information (Zebari *et al.* 2020).

In *Figure 2. 2*, the image containing a face mask examines certain regions of the image at a time. Once the region being examined is determined, the pixel values are further investigated from that region of the image (Bhatt *et al.* 2021). As illustrated, the convolution process is elucidated through the following representation which include the input image that is represented by the letter *I* and the filter represented by the letter *f*, forming an expression using the following computation pattern as depicted in *equation 2.1.*

$$Z = I * f \tag{2.1}$$

Within *Figure 2. 2*, a 3x3 input image is used to illustrate the computation process of the filter. The filter passes through a significantly small portion of the input image at a time.

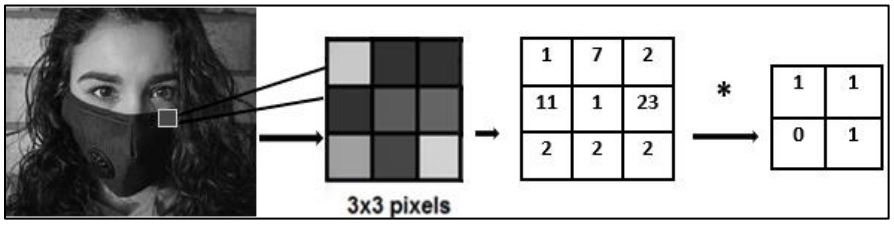


Figure 2. 2: CNN Filter Application to a Basic Input Image.

In the subsequent subprocess, the convolutional layer formula elucidates the mathematical narrative behind the convolution operation.

2.1.2.1. Convolutional Layer Formula

Equation 2.2 provides a depiction of the formula used to calculate one pixel in the subsequent layer during the convolution operation.

$$net(t,f) = (x * w)[t,f] = \sum_m \sum_n x[m,n] \cdot w\,[t-m],[f-n] \ , \tag{2.2}$$

where the output in the subsequent layer can be represented by *net(t, f)*. The input image is represented by *x* and the filter can be represented as *w*. Lastly the convolution operation can be represented by an *.

This operation involves the process of sliding the filter w across the input image represented by the letter x. For each position (t, f) in the output, an element-wise multiplication is performed between the filter and the corresponding region of the input image, followed by summing the results. The sums produce the value at position (t, f) in the output feature map. The summations iterate over the respective filter's dimensions, with indices m and n spanning the kernel's width and height, respectively. This operation helps to extract pertinent features such as edges, textures, and patterns from the input image (Albawi, Mohammed and Al-Zawi 2017).

In *Figure 2. 3*, within the context of the convolution process, a filter is applied to a matrix as represented in the second column of blocks. This process involves multiplication of the filter values with corresponding values represented by a different block colour (i.e., blue, orange, green and yellow) within the matrix. It is crucial to emphasise that in the second row (i.e., the matrix represented by the colour *orange*), the filter overlaps with the first region (i.e., the region from the first row represented by the colour *blue*) to ensure all regions of the image have been accounted for, processed, and are considered in the analysis. Hence, in the *orange* matrix, the filter passes through the first and second region of the image. The products of these multiplications are then summed up to produce a single value, demonstrating the result of the convolution operation which is represented by the single value in the last column (i.e., the single block value). The process is repeated uniformly across the entirety of the image, resulting in a new matrix of convolved values. Finally, the convolved matrix serves as the output, highlighting significant features extracted from the original image through the convolution operation.

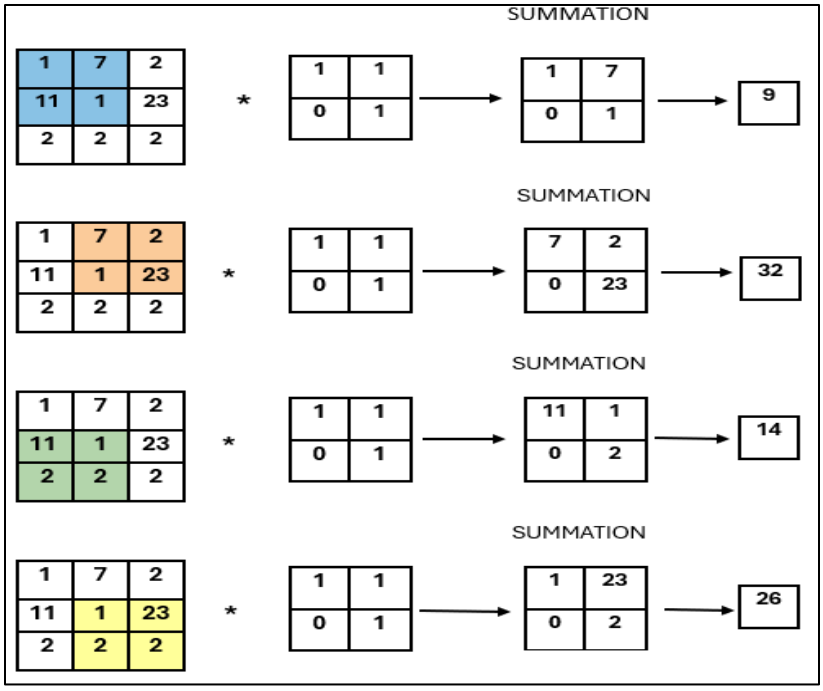


Figure 2. 3: The Main Computations Performed at Each Stage of the Convolutional Layer.

Furthermore, in this computation process the feature extraction layer is performed. The feature extraction layer refers to the process in which the curves and sharp edges of an image are extracted. Data (e.g., images) that is arbitrary in nature is transformed into numerical features that are utilised in the CNN (Jogin *et al.* 2018; Varshni *et al.* 2019).

In contrast to the preceding example highlighted in *Figure 2. 3*, where the input image underwent no implementation of the padding process, and a stride of one was employed, this indicates the selected stride across both vertical and horizontal positions within the kernel. It is worth noting that alternative stride values can be employed dependant on the task. Consequently, adjusting the stride value leads to the derivation of feature maps with reduced dimensions (Ahlawat *et al.* 2020). The padding and stride process can be further described in preceding subprocesses below.

2.1.2.2. Padding Process

The padding subprocess occurs within the input image (Albawi, Mohammed and Al-Zawi 2017). When a convolution computation process is performed, the input image size effectively shrinks, thus every time a convolution operation occurs the image is reduced in size (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). Hence, padding is applied to ensure spatial

dimensions are maintained whilst enabling improved feature extraction and preserving information integrity within the image (Naseri and Mehrdad 2023). In the convolution step, information on the boarder of an image is frequently compromised or lost. This occurs due to the fact that the filter only slides once when encapsulating the edges of an image (Ahlawat *et al.* 2020). For example, considering the application of a filter that contains a width of three and a length of three (i.e., a 3x3 image) that is applied to an 8x8 image with a stride of one, the filter only passes once as depicted in *Figure 2. 4*.

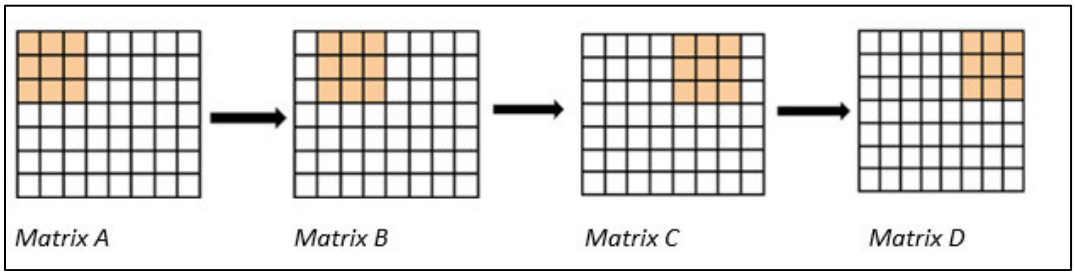


Figure 2. 4: *Padding Process*

In *Figure 2. 4*, both blocks in *Matrix A* and *Matrix D* are right at the edge of the image. Hence, the boarders of the image are not contained and trained as effectively as the ones contained in *Matrix B* and *C*. Padding is therefore the process of adding layers to the edge of the matrix once a convolution computation has occurred.

This essentially implies that when the new convolution is performed a padding layer of zero around the edges of the matrix is added to all four sides of an image (Naseri and Mehrdad 2023). This brings the image back to an 8x8 image, making it the same size of the original image. The image therefore allows for the boarder of the image to be preserved, and the filter can identify the edge as the middle of the respective image. Hence, a more thorough computation can take place (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). *Figure 2.* 5 displays an input image with a padding of zero around the boarders along with a filter of 3x3.

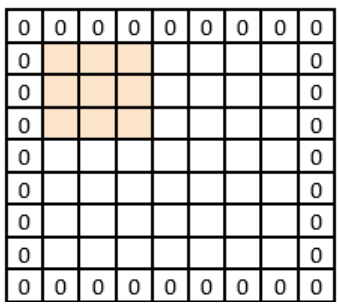


Figure 2. 5: Matrix with a Padding of Zero.

Using *equation 2.3*, the output formula with padding can be identified as:

$$O = 1 + \frac{N+2P-F}{S} \quad , \tag{2.3}$$

Where *O* denotes the output size, and *N*, *F*, and *S* respectively represent the input size, filter size, and stride. The letter *P* represents the number of layers in which a padding of zero is applied to the computation process. The padding function is used to adjust the input dimensions, ensuring that the convolutional filter can fully process edge elements of the input data (Albawi, Mohammed and Al-Zawi 2017). After the successful completion of the padding process, the subsequent subprocess known as the stride is applied.

2.1.2.3. Stride Process

During the convolutional process, the size of a stride needs to be agreed upon and applied to the convolution operation. A stride refers to the process of the step size required and applied when placing the filter over the input image (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). The size of the stride is dependent on the amount of information that can be gathered regarding the initial input image required for the feature map process. Hence, the smaller the size of the stride, the more in-depth information is gained from each region of the image (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). A simple analysis of the regions reveals there are many overlaps between the nodes in the next layer and their neighbours. Therefore, controlling the stride can assist in manipulating the overlap of the regions from the input image (Albawi, Mohammed and Al-Zawi 2017; Kong and Lucey 2017). Since a large portion of the filter overlaps the input image, a greater amount of features will be shared amongst the output (Yepez and Ko 2020). *Equation 2.4*, is utilised to calculate and formally present the information noted above. Using $N \times N$ which refers to the dimension of the input image and $F \times F$ which refers to the filter size dimension, the following equation can be formally introduced.

$$O = 1 + \frac{N-F}{S} \quad , \tag{2.4}$$

where *N* represents the input image size, *F* refers to the size of the filter, *S* represents the stride size, and *O* refers to the output size. Once the stride process has been successfully completed, the next step to be conducted is known as feature and weight sharing.

2.1.2.4. Feature and Weight Sharing

To observe and study the pixels identified in the kernels and to furthermore develop a visualisation needed to essentially classify an image, weight sharing is required. This process is often supported by convolution layers (Naseri and Mehrdad 2023). During the computation of the CNN process, weights are often attached to a neighbouring pixel. This process is carried out to ensure that when an image is being processed, the features are extracted from every part of the image (Jogin *et al.* 2018; Naseri and Mehrdad 2023). In the computation process of the CNN, there are many invariance transformations that take place when weight sharing occurs (Albawi, Mohammed and Al-Zawi 2017). Irrespective of the 3D properties, it can be used to help filter the learning features. In the process of learning the edge, the process commences with random values for the filters to determine if it improves the performance once the computation has been completed. It is important to note that utilising a shared weight is a commonly encountered error that occurs when endeavouring to ascertain the spatial significance of the input (Albawi, Mohammed and Al-Zawi 2017; Jogin *et al.* 2018; Naseri and Mehrdad 2023). Therefore, it is imperative that careful consideration be given to the design of the CNN to ensure appropriate feature and weight sharing strategies are implemented for optimal performance. Upon the completion of feature and weight sharing in CNN, the pooling layer is applied.

2.1.3. Pooling Layer

Advancing further, the pooling layer and the flattening process, which are crucial components necessary to the functioning of the CNN are described in detail below.

2.1.3.1. Pooling Layer Process

The purpose of the pooling layer function is to provide a simple layout to reduce the complexity of the layers. Hence, a down sample occurs. The pooling layer process represents the down sampling of features in an image to create a summation of information critical to identifying the image in the most efficient way possible (Sun *et al.* 2017). Therefore, the spatial volume of an image can be reduced through down sampling (Naseri and Mehrdad 2023). There are many types of pooling techniques that can be applied, including the minimum, maximum, average, and global average pooling, as depicted in *Figure 2.* 6. Among these different types of pooling methods, the average and maximum pooling are most commonly utilised methods (Akhtar and Ragavendran 2020). The first technique refers to the average pooling technique where the sub-region rectangles

of an image are summed up in the particular region and are then expressed as a fraction over the total number of pixels in that specific sub-region (Sun *et al.* 2017; Naseri and Mehrdad 2023). The average value is then taken and placed into a new matrix. This computation continues until all regions have been successfully summed and an averaged is calculated (Akhtar and Ragavendran 2020).

Figure 2. 6: Pooling Layer Summation

The most common strategy is known as max pooling. This pooling occurs whereby the image can be partitioned into multiple sub-regions that are in a rectangular format. Based on the specific sub-region, the maximum value is taken and returned to form a new matrix. This computation process continues until all sub-regions have been identified (Albawi, Mohammed and Al-Zawi 2017; Akhtar and Ragavendran 2020; Naseri and Mehrdad 2023). In this pooling technique the 2x2 size is often utilised as a filter which is slid across the matrix. As seen in *Figure 2. 6*, under the *max pooling* heading, given an input image where a convolution operation has been performed, the new matrix can be seen as a 4x4 input. The pooling process initiates from the top left corner and traverses through the image in two-by-two blocks, utilising a stride of two. This continues until it reaches the top right corner. The process therefore continues until all four sub-regions have been taken into consideration and the maximum value is identified for each sub-region (Albawi, Mohammed and Al-Zawi 2017). Using *equation 2.5,* the max pooling layer is formally presented below.

$$P_{j,m} = max\big(h_j, (m-1)N + r\big) \; for \; r \in \{1, \dots, R\} \tag{2.5}$$

In this operation, the feature map is essentially divided into non-overlapping and over-lapping regions the size of R. Based on these regions the maximum value is selected from each region. For each pooled element denoted by $P_{j,m}$, the maximum value contained in a local region h_j is determined by evaluating all activations in the region defined by $r \in \{1, \dots, R\}$ within the region of

size R. The index j is used to represent a specific activation within the specified region. The letter m is representative of the position of the pooling region across the feature map presented. The stride N controls the step size between consecutive pooling operations. This stride affects the amount of overlap between the pooling regions present. The output feature map's size which is denoted as M, is determined by the stride N and the pooling region size R, as depicted by the formula $M = \frac{K-R}{N} + 1$, where K is the number of input feature maps (Akhtar and Ragavendran 2020; Gholamalinezhad and Khosravi 2020; Alzubaidi *et al.* 2021). Upon completion of the pooling layer computation process, the pooled features then need to be flattened. This subprocess can be conducted through the utilisation of a technique titled flattening.

2.1.3.2. Flattening Process

Flattening refers to the transformation of a multi-dimensional array into a singular, one-dimensional (1D) array. The 1D array is then fed into the classification model. In the flattening process, data is divided into smaller regions, thus, the smaller dimension in turn helps to ensure the training process is swift (Albawi, Mohammed and Al-Zawi 2017; Ayyadevara 2018; Naseri and Mehrdad 2023). Since the data is stored in a single dimension there is also a reduction in the memory requirements needed to provide quick computation for the CNN model. This process is conducted for the purpose of ensuring that the model can correctly be fed to the CNN model (Ashin *et al.* 2021). *Figure 2. 7* highlights the computational process of converting the multi-dimensional array or pooled feature maps into a single dimension or single long vector of input data. The flattened layer is then connected to the fully connected layer.

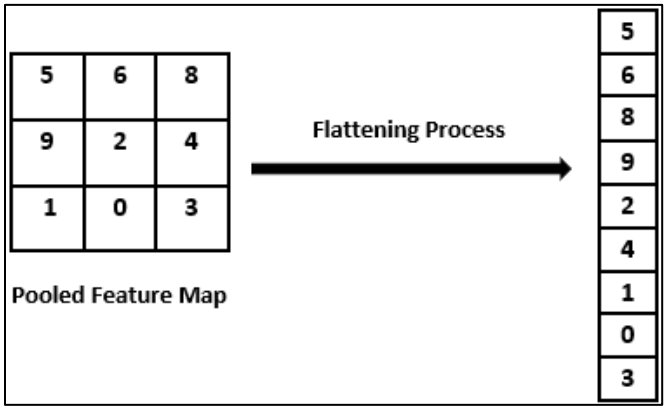


Figure 2. 7 : Flattening Process

2.1.4. Fully Connected Layer

Within the CNN architecture, the activation function that lies within the fully connected layer process, plays a crucial role, serving to introduce non-linearity to the network's computations. Following this, the fully connected layer, incorporating the activation function, further refines extracted features for accurate classification or regression tasks.

2.1.4.1. Activation Function

Initiating the process of non-linearity requires the implementation of the activation function. The non-linearity process is introduced into the neuron's output (Naseri and Mehrdad 2023). The purpose of introducing non-linearity is to simply remove certain regions to limit the output generated. The activation function can be utilised as a tool to assist in determining whether a single neuron in the model should be activated or not (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). The steps required to calculate whether a neuron should be activated or not follows the pattern highlighted in *equation 2.6*.

$$Output = sum\,(input * weight) + bias. \tag{2.6}$$

Within this equation, the *weight* represents how effective a particular input is. The steepness of the activation function can therefore be represented by a *weight*. The *input* represents the input image pixel value. The *bias* can be described as features and weights which are multiplied by this constant. The constant is known as the *bias* (Ertuğrul 2018; Naseri and Mehrdad 2023). This non-linear transformation is conducted on the input. Hence, the model is more efficient and capable when learning and performing tasks that are complex in nature (Agarap 2018; Hao *et al.* 2020). The most widely employed non-linear activation functions utilised within deep neural networks include the Rectified Linear Unit (ReLU), Leaky ReLU, Noisy ReLU, Tanh and Sigmoid (Alzubaidi *et al.* 2021)**.** Within the most common activation functions**, t**he most popular activation functions include the ReLU, Tanh and Sigmoid.

I. Sigmoid Function

The Sigmoid function, which is also referred to as the logistic function, is utilised to transform input values into a range between 0 and 1 (Bhatt et al. 2021; Kaloev and Krastev 2021). This inherently resembles a characteristic within the S-shaped curve. The sigmoid function is significantly useful for binary classification tasks, where it fits input values to

represent probabilities, thus, aiding in decision-making processes (Naseri and Mehrdad 2023). The sigmoid function can be represented using *equation 2.7* (Kaloev and Krastev 2021).

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1} = 1 - S(-x) \tag{2.7}$$

II. Tanh Function

Through the utilisation of the Tanh function, input values between a range of -1 and 1 are mapped (Alzu'bi *et al.* 2021; Naseri and Mehrdad 2023). This is conducted in order to allow for the modelling of complex relationships between the specified features (Alzubaidi *et al.* 2021). The tanh function's S-shaped curve makes it particularly effective in capturing both positive and negative aspects of data (Kaloev and Krastev 2021). This enables improved decision-making in classification and regression tasks within the field of machine and deep learning. The tanh function can be represented using *equation 2.8* (Kaloev and Krastev 2021).

$$tanh\, x = \frac{sinh\, x}{cosh\, x} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1} \tag{2.8}$$

III. ReLU Function

The ReLU activation function is popular in nature due to its fast computation speed (Ertuğrul 2018; Kirana *et al.* 2019; Naseri and Mehrdad 2023). Both the function and gradient classifications of ReLU are simpler in nature (Ertuğrul 2018). The ReLU activation function output is illustrated in *Figure 2. 8.*

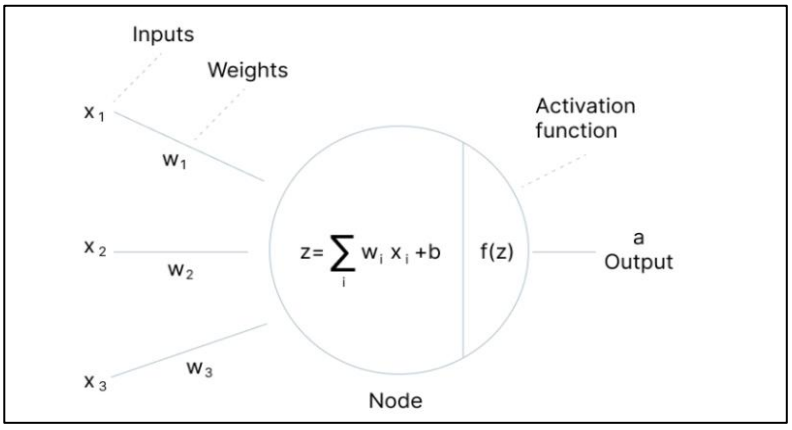


Figure 2. 8: ReLU Activation Function Output (Shams 2023)

During back propagation the sigmoid and tanh function often cause problems such as the vanishing gradient problem. This problem often hinders the training process causing the gradients to approach a value of zero as they propagate through the multiple layers within this process (Naseri and Mehrdad 2023). However, if the ReLU function is utilised, this function has a gradient that will always be constant when a positive input is given. It is possible to ignore the difference of the function in the implementation even though it is not differentiable. As a result of the zero in the gradient, the ReLU function thus creates a sparser representation (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023). The function can be represented by *equation 2. 9*.

$$f(x) = max(0, x) \; and \; \frac{d}{dx} relu(x) = 1 \; if \; x > 0; \; otherwise \; 0 \tag{2.9}$$

Following the activation function, the introduction based on the implementation of the fully connected layer is described in detail.

2.1.4.2. Implementation of the Fully Connected Layer

The fully connected layer is connected to the end of the network. This layer consists of a finite number of neurons that receive an input in the form of a vector. Subsequently, the input undergoes transformation through a weights matrix, yielding another vector as the output (Naseri and Mehrdad 2023). Based on the extracted features derived from the earlier stages, it predicts the image's class based on the output of the convolution process. Therefore, a direct connection exists between each node in the preceding layers and the subsequent layers of the model (Albawi, Mohammed and Al-Zawi 2017). Hence, the fully connected layer presents the link between the connected layers and the output. This further reinforces that an input vector inherently influences an output vector within the model. This layer receives an input based on the output of the convolution and pooling layer and is then flattened. Once flattened, the final input with the respective weights and biases are fed into the fully connected layer of the model (Albawi, Mohammed and Al-Zawi 2017; Naseri and Mehrdad 2023).

This layer is inherently used to understand and learn non-linear feature combinations or functions. When the model processes a neuron, a linear transformation is applied utilising a weights matrix on the input vector (Naseri and Mehrdad 2023). Once the linear transformation has been applied a non-linear transformation can then be applied through an activation function *f* that is non-linear in

nature to the product (Albawi, Mohammed and Al-Zawi 2017). *Figure 2. 9* displays a realistic view of the CNN process. This process considers a real input image and displays the five step CNN architecture for the computation of the neural network to perform an accurate classification based on the type of animal class the algorithm predicts the input image belongs to.

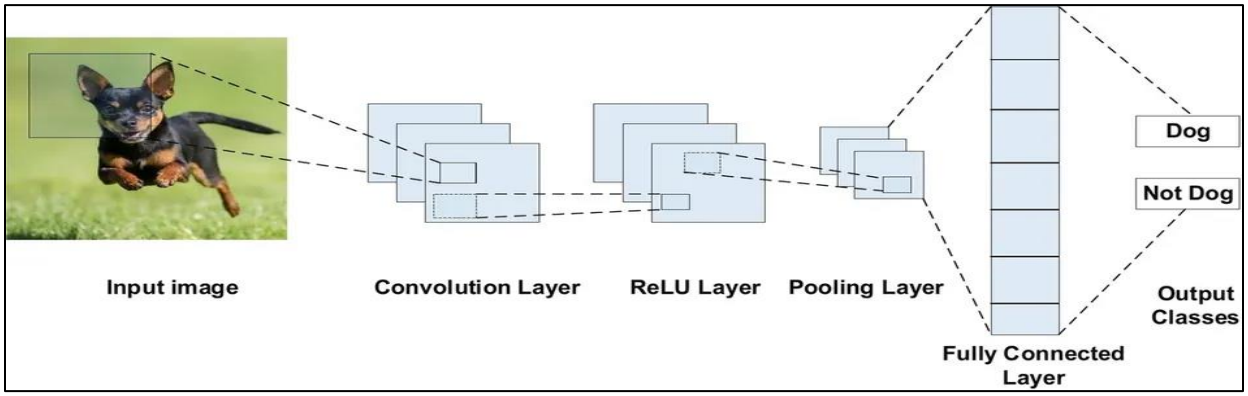


Figure 2. 9: Fully Connected Layer (Staff 2021)

2.1.5. Output Layer

The last layer better known as the output layer utilises the one-hot encoding method to encode the output labels expected by the algorithm to generate (Alzubaidi *et al.* 2021). One-hot encoding is a method that can be identified as a technique which is utilised to present different categorical variables as a numerical value during the execution of the CNN model at run-time. Therefore, more information can be presented based on the categorical variable which in turn improves the CNN model drastically (Bagui *et al.* 2021; Naseri and Mehrdad 2023).

2.2. Architectural Approaches in Face Mask Detection

Within the realm of the CNN, MobileNetV2 has been identified as a light weight CNN architecture that consists of a unique blend of efficiency and accuracy making it a suitable choice for the implementation of FMD (Al-Rammahi 2022; Hussain *et al.* 2022). Due to its lightweight architecture and efficient utilisation of computational resources, MobileNetV2 is a respectable choice for mobile and embedded devices making real-time detection easy and efficient (Rokhana, Herulambang and Indraswari 2021). This architectural design is particularly efficient in resource-limited environments, due to its utilisation of depth-wise separable convolutions, which significantly decrease computational demands (Sandler *et al.* 2018). This architecture is capable of providing robust performance under a variety of conditions and demographics by effectively

capturing intricate features that are essential for accurate mask detection (Hussain *et al.* 2022; Kumar and Bansal 2023). Within the context of MobileNetV2, efficient inference speed enables swift processing of image data, thus facilitating the rapid identification of mask-wearing behaviour in a dynamic environment that the model is presented with (Nagrath *et al.* 2021). Furthermore, MobileNetV2 provides competitive accuracy in image classification tasks, making it possible to accurately identify individuals wearing or not wearing masks, which is crucial to public safety (Al-Rammahi 2022). Studies conducted by Christa *et al.* (2021); Nagrath *et al.* (2021); Hussain *et al.* (2022); Kumar and Bansal (2023) have proven that the MobileNetV2 architecture is capable of producing respectable accuracy rates for image classification in FMD.

2.2.1. MobileNetV2 Architecture

The architectural foundation of MobileNetV2 rests within the foundation of the initial input layer where the images undergo a series of preprocessing techniques before it is fed into the subsequent layers (Sandler *et al.* 2018). The backbone of the MobileNetV2 architecture relies on the depth-wise separable convolutions. Depth-wise separable convolutions are utilised for the purpose of enabling significant reductions in computational costs whilst maintaining valuable feature information required for computation (Sandler *et al.* 2018; Yong *et al.* 2023). In addition, bottleneck structures are applied to reduce information loss through the application of linear activation functions as opposed to the commonly utilised non-linear functions (Sandler *et al.* 2018; Yong *et al.* 2023). A sequence of inverted residual blocks are followed thereafter to enhance the model's accuracy through the incorporation of a bottleneck structure (Sandler *et al.* 2018). In order to simplify classification, the global average pooling layer is applied to reduce the number of feature maps to a practical size (Sandler *et al.* 2018; Yong *et al.* 2023). The output layer is utilised to synthesise the network's predictions in order to produce actionable insights or classifications based on the input data provided (Yong *et al.* 2023). Hence, the MobileNetV2 architecture consists of the following layers:

- Input Layer
- Depth-Wise Separable Convolutions
- Linear Bottlenecks
- Inverted Residual Blocks
- Global Average Pooling Layer

- Output Layer

2.2.1.1. Input Layer

The process begins with the input layer. The input layer is utilised as an initial access point for the data to be fed into the neural network architecture. The input layer can be mathematically denoted using x_{in}, where the letter x used to represent the new data based on the specified datasets that are input into the model while *in* is used to represent the designated input layer of the network (Sandler *et al.* 2018; Dong *et al.* 2020). For example, in an image classification task such as FMD, an input image with dimensions of 160x160 and a greyscale channel of one, a matrix of 160x160x1 is used to represent the pixel values of the input image. Once the input layer transformations have been applied to the respective images, the subsequent process of depth-wise separable convolutions are applied (Yong *et al.* 2023).

2.2.1.2. Depth-Wise Separable Convolutions

The depth-wise separable convolution layer is made up of the two different operations which include the depth-wise and point-wise convolutions as depicted in *Figure 2. 10* (Lian, Wang and Zhang 2022). Within the context of the point-wise convolution operation, information is inherently aggregated across channels utilising convolutions of 1x1 (Sandler *et al.* 2018). This ensures the preservation and retention of valuable features (Yong *et al.* 2023). In depth-wise convolutions, computation complexity is reduced since each input channel is convolved independently whilst maintaining its own filter. In order to understand the mathematical application utilised in the depth-wise separable convolution, *equation 2.10* is formally presented.

$$Y = PointwiseConv(DepthwiseConv(X, W_d), W_p\ ), \tag{2.10}$$

where the output feature map Y is derived by combining the results of the depth-wise convolution with the point-wise convolution. The input feature map is represented by the letter X. Using W_d the depth-wise convolution kernels are represented and, Wp represents the point-wise convolution kernels, which are applied as a multiplication to the output of the depth-wise convolution to combine the channels. The second comma serves to delineate the two distinct operations which include the depth-wise convolution, this is promptly followed by point-wise convolution, emphasising the sequential flow of data between them (Sandler *et al.* 2018; Lian, Wang and Zhang 2022). Upon the depth-wise separable convolution operation, the linear bottleneck process is applied.

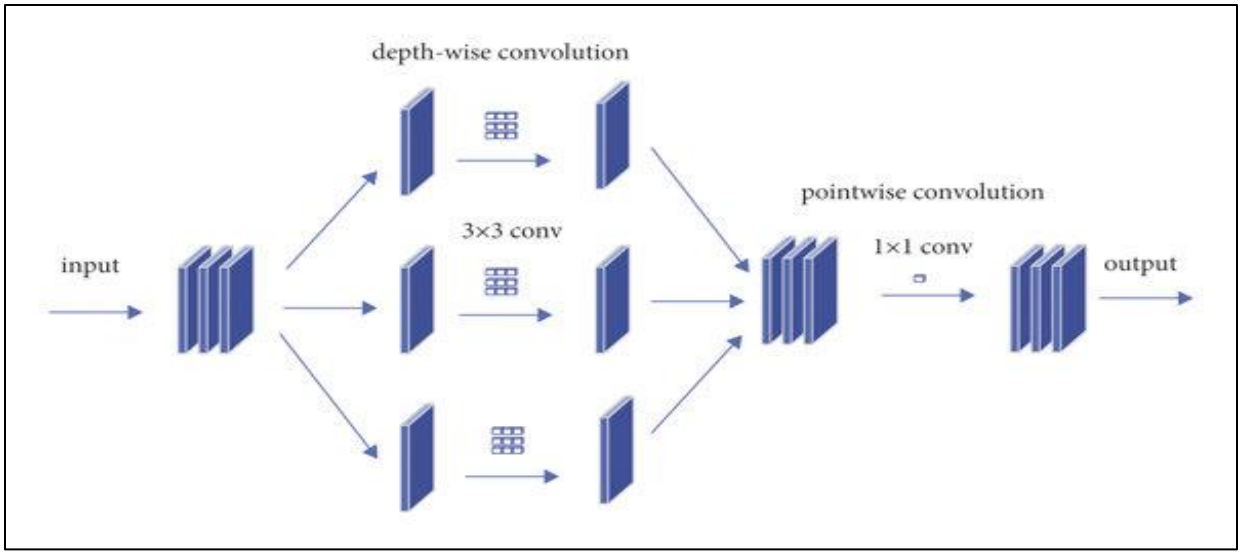


Figure 2. 10: Depth-Wise Separable Convolution Operation (Lian, Wang and Zhang 2022)

2.2.1.3. Linear Bottlenecks

Since MobileNetV2 utilises a linear bottleneck process, identity mappings can be easily accessed through the employment of shortcut connections that reside within the architecture (Khandelwal *et al.* 2023). The network is easily converged since the model possesses the ability to learn identity mappings when required (Sandler *et al.* 2018). Linear bottlenecks are pivotal for enhancing network capacity without increasing computational demands within the model. Hence, it constitutes a fundamental component of efficient deep learning architectures (Yong *et al.* 2023). This component coordinates a linear transformation of feature maps (Khandelwal *et al.* 2023). It then proceeds to follow the application of a non-linear activation function. This process is conducted by a commonly utilised non-linear ReLU activation function. To grasp the mathematical foundations of the linear bottleneck layer employed, *equation 2. 11* is utilised (Sandler *et al.* 2018; Dan, Sun and Liu 2019; Yong *et al.* 2023).

$$Y = ReLU(Conv(X, W) + b) \quad (2.\ 11)$$

X refers to the input feature map and the convolutional kernel is represented by *W*. The letter *b* is used to refer to the bias and the non-linear activation function is denoted by the ReLU term (Sandler *et al.* 2018; Yong *et al.* 2023). The next process following the linear bottlenecks is known as the inverted residual block process.

2.2.1.4. Inverted Residual Blocks

Depicted in *Figure 2. 11,* inverted residual blocks are used to provide more detail and to distinguish between different classes or categories in the data. The inverted residual blocks enhance the ability of the network to capture subtle variations and complex patterns that reside within the data (Dan, Sun and Liu 2019; Yong *et al.* 2023). This process is conducted through the incorporation of skip connections. The purpose of a skip connection is its ability to enable gradient flow during training and improve the vanishing gradient problem by directly connecting inputs from one layer to outputs from a future layer (Sandler *et al.* 2018). Thus, the network can easily leverage the learned features and efficiently reproduce information across the various layers of the MobilNetV2 network (Yong *et al.* 2023). Embedded within these blocks is a streamlined bottleneck. This effectively provides a way for subsequent point-wise convolutions and another layer of linear bottlenecks. This configuration promotes an optimised flow of information through the network. Using *equation 2. 12,* the residual inverted block process is described (Sandler *et al.* 2018; Lian, Wang and Zhang 2022; Yong *et al.* 2023).

$$Y = F_{id}\left(F_{expand}(X)\right) + X \tag{2. 12}$$

The F_{id} is used to represent the depth-wise separable convolutional layer. The letter X is used to define the input feature map. The expansion layer is denoted by F_{expand}. In the last part of the equation, the $+X$ represents the skip connection summation applied (Lian, Wang and Zhang 2022; Yong *et al.* 2023).

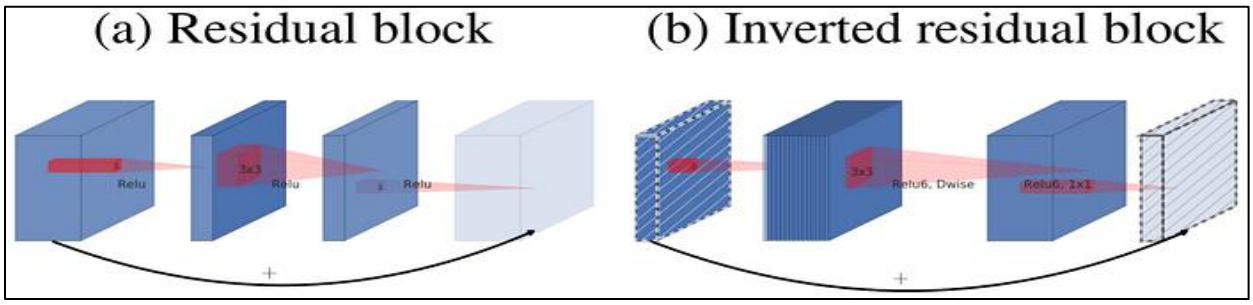


Figure 2. 11: Inverted Residual Block Operation (Sandler et al. 2018)

Within the main inverted residual block framework, additional subprocesses are utilised to strengthen the network performance effectively. These include the incorporation of the expansion layer, width multiplier and the residual connections. These three subprocesses are used to collectively enhance the efficiency and effectiveness of the model.

I. Expansion Layer

The purpose of the expansion layer is to effectively allow for the computation of complex transformations to be applied whilst enhancing the network capacity (Lian, Wang and Zhang 2022). Therefore, it increases the dimensionality of feature maps (Sandler *et al.* 2018). This expansion is conducted in the expansion layer to promote the capturing of intricate patterns in the data. With reference to *equation 2.13,*

$$F_{expand}(X) = Conv(X, W_{expand}) \quad (2.13)$$

X is used to represent the input feature map. The expansion layer weights are defined by W_{expand} and the convolution operation is denoted by *Conv.* This is used to apply a set of learnable filters denoted by the W_{expand}, to the input feature map X to generate the expanded feature map $F_{expand}(X)$ (Sandler *et al.* 2018). Within the expansion layer, the implementation of the width multiplier is applied as described below.

II. Width Multiplier

An expansion layer of inverted residual blocks is adjusted by a width multiplier parameter. Using the width multiplier, the model's width or computational cost can be controlled through the process of scaling the number of channels (Chin *et al.* 2020). MobileNetV2 is able to achieve a balance between model performance and computational efficiency through this parameter. As an example, when employing a width multiplier of 0.5, the quantity of the channels are divided in half. This reduction effectively lowers computational demands while preserving the model's capacity to a specified degree (Lian, Wang and Zhang 2022; Yong *et al.* 2023).

III. Residual Connections

Residual connections are implemented within the inverted residual block operation in order to facilitate the flow of information whilst aiding in gradient propagation that occurs during the training process (Sandler *et al.* 2018). This subprocess is used to establish direct connections between the input and output feature maps presented to the model. Residual connections serve to lessen the challenge of the vanishing gradient problem by fostering a training environment that is both stable and conducive to the efficient optimisation of deep neural networks (Lian, Wang and Zhang 2022). *Equation 2.14* outlines the operational workings of the residual connections, uncovering how it dynamically fine-tunes feature responses.

$$Y_{output} = F_{id}\left(F_{expand}(X)\right) + X \quad (2.14)$$

The Y_{output} defines the output feature map. This is used to obtain the depth-wise separable convolution represented by F_{id}. Hence, this is applied to the expanded feature map that is denoted by the $F_{expand}(X)$. The X at the end of the equation is added to the residual connection. The X is used to represent the input feature map (Meng *et al.* 2021).

The architecture then proceeds to the next layer better known as the global average pooling layer.

2.2.1.5. Global Average Pooling Layer

Within the neural network architecture, the global average pooling layer assumes a critical role, especially in tasks such as image classification. Its primary function involves the combination of spatial information from all feature maps (Sandler *et al.* 2018; Dan, Sun and Liu 2019; Yong *et al.* 2023). This amalgamation process serves to compress the spatial dimensions into a singular vector. Hence, it can effectively capture the core information of the features that are derived from the previous layers (Üzen *et al.* 2023). Using *equation 2.15,* the global average pooling layer can be formally represented.

$$Y_i = \frac{1}{h \times w} \sum_{j=1}^{h} \sum_{k=1}^{w} X_{i,j,k} \quad (2.15)$$

The computation of the output Y_i for each feature map i within the global average pooling layer follows a systematic procedure. Initially all elements are aggregated within the i^{th} feature map by summing them up. Mathematically, this summation is represented as $\sum_{j=1}^{h} \sum_{k=1}^{w} X_{i,j,k}$. Here, the $X_{i,j,k}$ denotes the value of the i, j, k^{th} element in the feature map. Next, normalisation occurs by normalising the sum by dividing it by the total number of elements in the feature map, which is the product of its height h and width w, to form the fraction $\frac{1}{h \times w}$. This normalisation ensures that the resulting output remains independent of the spatial dimensions of the feature map (Sandler *et al.* 2018; Dan, Sun and Liu 2019; Yong *et al.* 2023).

Upon completion of the global average pooling layer, the feature maps go through a transformative process where they are compressed into a singular vector. This transformation streamlines the transition to the subsequent layers, often fully connected, which are integral for classification tasks (Sandler *et al.* 2018). This compression mitigates the complexity of the data. In addition, it is also responsible for safeguarding pertinent information necessary for pattern recognition. The global

average pooling layer's capacity to condense spatial complexities into a concise format highlights its essential role within neural network structure (Lian, Wang and Zhang 2022). This inclusion improves the performance and operational efficiency of the network across the FMD domain (Lian, Wang and Zhang 2022; Yong *et al.* 2023). Once the global average pooling layer has been conducted, the Squeeze and Excitation (SE) blocks that are utilised to adaptively fine-tune the channel-wise features are introduced.

I. Squeeze and Excitation Blocks

Embedded within each inverted residual block, SE blocks contribute to augmenting the network's capacity for representation (Üzen *et al.* 2023). Through dynamic adjustment of channel-specific feature responses, SE blocks emphasise important spatial features while reducing the importance of the less relevant identified features. This enables MobileNetV2 to concentrate on discriminative features, thereby enhancing its efficacy across diverse tasks such as image classification and object detection (Yong *et al.* 2023). Using *equation 2.16,* the SE block operation can be further described.

$$Y_{se} = \sigma\big(W_2\ \delta(W_1\ \bar{X})\big) \tag{2.16}$$

The global average pooling symbol can be denoted by $\bar{X}$ for the input feature map of the model. Subsequently, the weights of the two fully connected layers are represented by the letters W_1 and W_2. Denoted by δ, the ReLU activation function is applied. The σ is used to represent sigmoid activation function in this model (Meng *et al.* 2021; Cao *et al.* 2023). Upon the completion of the global average pooling layer and the SE block implementation, the architecture proceeds to the next layer better known as the output layer.

2.2.1.6. Output Layer

Based on the extracted features collected from the input data that is fed into the model, the network produces a prediction forming the output layer (Sandler *et al.* 2018). In the output layer, each neuron inherently represents a class label for classification in FMD. Within the realm of image classification for FMD, the output layer is responsible for producing the probability distribution over the classes (Yong *et al.* 2023). This is conducted in order to indicate the likelihood of the specified image belonging to each class within the respective categories. Through the implementation of *equation 2.17*, the output layer is defined in detail.

$$Y_{output} = Softmax\ (W * X + b) \quad (2.17)$$

The Y_{output} defines the final outcome based on the respective classification. Within context of W, the weight matrix is represented. Softmax refers to the specified activation function utilised by the model. The X which represents the input feature map is multiplied by the weight matrix. By adding the bias, which is defined by the b symbol, the final classification is performed (Lian, Wang and Zhang 2022; Cao *et al.* 2023; Üzen *et al.* 2023).

2.3. Architectural Approaches in Masked Facial Recognition

Navigating through the complex challenges posed by MFR requires the vital development of a robust architecture. The proposed model, composed of a host of different yet significant architectures are incorporated to develop a comprehensive model for MFR. Knowledge of each component's architecture is crucial to understand toward the development of the MFR model. The Multi-Task Cascaded Convolutional Neural Network (MTCNN), utilised for detection, cropping and landmark detection of the facial region serves as the foundation for the MFR model (Chunming and Ying 2021). Once the MTCNN detection and facial region cropping process has been completed, key landmark points are drawn on the located face using MTCNN and OpenCV to aid in the next step of feature extraction. The FaceNet InceptionResNetV1 model emerges as a powerhouse architecture for extracting discriminative features based on the detected facial landmarks retrieved from MTCNN (Kheaksong *et al.* 2022). Driven by the Euclidean Distance metric, the similarity of the features are ascertained based on the calculation of closeness. This facilitates the process of efficient face matching within the model (Kortli *et al.* 2020). The process concludes with the storage of numerical features and utilises the CNN architecture to successfully train the model for accurate classification of occluded faces. Hence, the architecture utilised for each of these components are discussed in detail. The main components within the MFR model architecture include:

- Multi-Task Cascaded Convolutional Neural Network
- FaceNet InceptionResNetV1 Model
- Convolutional Neural Network Model (*Section 2.1*)

2.3.1. Multi-Task Cascaded Convolutional Neural Network Operation

The MTCNN is a face detection and localisation algorithm that is developed to detect and locate faces (Chunming and Ying 2021). Extending itself beyond simple face detection capabilities, the MTCNN can detect faces when presented with partial occlusions, different image sizes, facial expressions, varied lighting conditions and angles (Kaziakhmedov *et al.* 2019; Qi and Yang 2020; Chunming and Ying 2021). The model works on devices that provide low consumption power, making it lightweight and efficient during computational processing (Kavitha *et al.* 2022). The MTCNN performs three separate CNN stages. This includes the face detection, facial landmark detection and facial boundary prediction. These three stages are combined to form a single robust model (Chunming and Ying 2021). The MTCNN model is subdivided into three specific stages namely the P-Net, R-Net and the O-Net.

I. P-Net

The P-Net component known as the "Personal Network" is utilised in the first stage of MTCNN. This algorithm utilises a fully connected CNN to produce a set of candidate bounding boxes that may contain faces (Qi and Yang 2020; Kavitha *et al.* 2022). The algorithm further goes on to identify potential facial regions by scanning an input image at multiple scales. The structure of this architecture includes a bounding box window size of 12x12 which is utilised for the P-Net face detection. Detecting a face requires the implementation of forward propagation. This is a method used to generate a feature map (Chunming and Ying 2021). For instance, within the feature map, a 32-dimensional feature vector is often utilised to determine whether the bounding box has identified a specific facial image or not. Once the facial image is detected, the model predicts the exact coordinates of the bounding box. Once completed, the box is applied around the face. Upon application of the bounding box to the specified face, the model retrieves the corresponding region from the original input image. A Non-Maximum Suppression (NMS) technique is then applied to keep the bounding box with the maximum score, whilst removing all other bounding boxes that possess an irregular large overlapping area (Qi and Yang 2020; Kavitha *et al.* 2022). This process can be illustrated in *Figure 2. 12.* Using *equation 2.18*, P-Net can be further described.

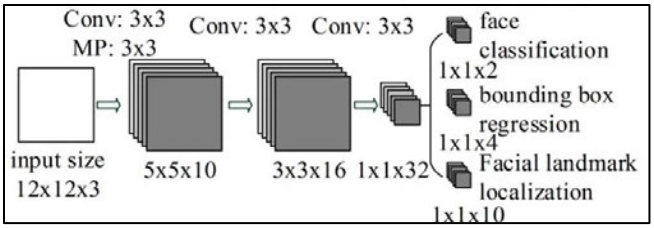


Figure 2. 12: P-Net Stage Process (Zhang, Luo and Gao 2020)

$$(b_i, c_i) = P\ net(F_{input}) \tag{2.18}$$

The variables b_i are used to represent the bounding box coordinates, while c_i represents, for each proposed face region, a specific confidence score. The F_{input} represents the input feature maps. In this equation, the bounding box coordinates and confidence scores for the facial regions are generated using the F_{input} as the input of P-Net. Based on their confidence scores, these bounding boxes indicate if a face is likely to be found in the input image (Zhang *et al.* 2018). The next subprocess then applied is known as the "Refine Network".

II. R-Net

The "Refine Network" (R-Net) is used to improve the detection of an identified facial region where R-Net identifies a false positive, the image is removed (Zhang, Luo and Gao 2020). The bounding box proposals from the P-Net stage are refined to perform accurate classification of facial and non-facial images. The R-Net is used to route P-Net candidates. Hence, R-Net serves as a pivotal candidate in the CNN architecture. Through the employment of bounding boxes, for example, a bounding box scaled to a size of 48x48 from the original 24x24 bounding box. Therefore, the accuracy of the bounding box detection and the facial landmark extraction is improved (Chunming and Ying 2021; Kavitha *et al.* 2022). The described R-Net architecture is therefore presented in *Figure 2. 13*. Using *equation 2.19*, the R-Net architecture is expressed.

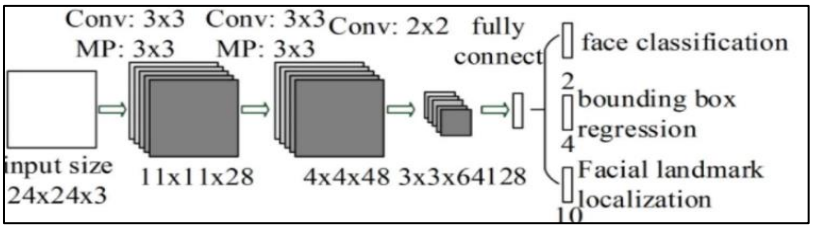


Figure 2. 13: R-Net Stage Process (Zhang, Luo and Gao 2020)

$$\Delta\, b_i = R\ net\ (F_{input}) \tag{2.19}$$

The Δ represents the adjustments required for the bounding box coordinates b_i that are inherently proposed in the P-Net stage. Therefore, bounding box coordinates proposed by the P-Net stage are refined using F_{input} as the feature maps, utilised by R-Net (Zhang *et al.* 2018). Following the application of the R-Net subprocess the next subprocess applied is known as the "Output Network".

III. <u>O-Net</u>

O-Net also referred to as the "Output Network", applies advanced pre-processed bounding boxes to the facial region for facial landmark detection. Facial landmarks, including but not limited to the eyes, nose, and mouth are identified to ensure feature extraction of the face is performed precisely with increased accuracy (Zhang, Luo and Gao 2020). For instance, by employing a method of approximation, a 12x12 box can be scaled to a 24x24 bounding box using P-Net. Subsequently, the data is then passed to the O Net. O-Net is therefore responsible for determining whether a face can be identified. The NMS technique is then applied. The NMS refers to a technique whereby the most relevant bounding boxes are selected whilst redundant or overlapping detections are filtered out to ensure that only the most confident and non-overlapping predictions are retained (Wang, Sun and Guo 2023). The NMS technique is therefore applied to the successfully detected human face once the bounding box regression method has been completed (Kakarla *et al.* 2020; Chunming and Ying 2021). The described O-Net architecture is presented in *Figure 2 14*. Using *equation 2.20*, the O-Net stage is further described.

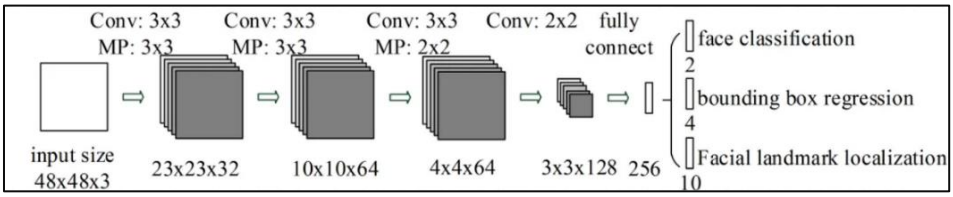


Figure 2. 14: O-Net Stage Process (Zhang, Luo and Gao 2020)

$$l_i = O\ net\ (F_{input}) \tag{2.20}$$

The variables l_i for each refined bounding box is represented by the facial landmark coordinates. Using R-Net as the input, the refined bounding boxes and the F_{input} for each face region, predicts the facial landmarks (Zhang *et al.* 2018).

Hence, through the utilisation of the three stages, the main MTCNN equation is formed as noted using *equation 2.21* (Zhang *et al.* 2018). The MTCNN model is used to combine the outputs received from each of the respective subprocesses to perform face detection, facial region cropping, and landmark detection capabilities as depicted in *Figure 2. 15.*

$$MTCNN\left(F_{input}\right) = O - net\left(R - net\left(P - net\left(F_{input}\right)\right)\right) \tag{2.21}$$

Figure 2. 15: MTCNN Face Detection and Cropping Function

Once the MTCNN has been applied, for face detection , facial cropping, and landmark detection, the next architectural process discussed lies within the FaceNet InceptionResNetV1 model architecture, building on the main architectural process for MFR.

2.3.2. FaceNet InceptionResNetV1 Architecture for Feature Extraction

Within the domain of the CNN, the FaceNet InceptionResNetV1 model has been selected as a respectable model for MFR feature extraction. FaceNet is designed for face recognition since it utilises high dimensional feature embeddings of the human face to extract features to perform facial recognition (Saleem *et al.* 2023). FaceNet applies the power of one-shot learning making it capable of producing respectable results (Cahyono, Wirawan and Rachmadi 2020; Kheaksong *et al.* 2022). Hence, the focus is on face embeddings which is also known as one-shot learning. One-shot learning is a novel loss function based on the Euclidean facial similarity measurement, which was introduced, abandoning traditional Softmax classification in favour of Triplet Loss as the new loss function. Therefore, this method utilises face embeddings to perform face matching on a particular person and then performs face recognition based on the similarity of the face embedding identified during the facial matching process (William *et al.* 2019; Jiang 2020). In addition, the model utilises a minimal number of training images. Thus, it reduces the need for computational

complexity and resource consumption during the training process (Darborg 2020). This model has therefore demonstrated its success in numerous research studies that have implemented FaceNet with CNN for face recognition (Sheikh *et al.* 2018; William *et al.* 2019; Yang, Ge and Zhang 2020; Wu and Zhang 2021).

The FaceNet InceptionResNetV1 architecture is a combination of the ResNet and Inception models. These models when combined produce a fully functional and cohesive model capable of providing excellent feature extraction for feature embeddings (Cao *et al.* 2024). The architecture is composed of the basic main components that follow a logical sequence of events to form the main overall architecture. Starting with the input layer which is used to extract the detected landmark points for each facial input image. Next the convolution layers are then used to analyse the relationships between the spatial features (Chunming and Ying 2021). Once the convolution operation is completed, in order to capture features across the relative scales, the inception modules are applied (Wu and Zhang 2021). The facilitation of gradient flow relies on the implementation of residual connections. In addition, the global average pooling technique is used to combine all the relevant information required to create consistent representations required for the model (Li, Deng and Chiang 2020; Deng *et al.* 2021). The model concludes with the output layer which is utilised to synthesise the network's predictions in order to produce actionable insights or classifications based on the input data provided. By controlling dimensionality precisely, the model ensures compact and discriminative facial representations that are crucial to accurate MFR (Cahyono, Wirawan and Rachmadi 2020). Therefore, the main components of FaceNet InceptionResNetV1 architecture for feature extraction include:

- Input Layer
- Convolutional Layers
- Inception Modules
- Residual Connections
- Global Average Pooling
- Output Layer

2.3.2.1. Input Layer

The input layer is used to receive the detected landmark points as the input for the feature extraction process. The purpose of this layer is to receive the spatial coordinates that are used to

represent the key facial features based on the identified landmark points from the MTCNN process (Wu and Zhang 2021). Hence, using the input layer as an entry point to the model, the input is denoted by the X_{input}, where the letter X is used to denote the input values inputted into the model (Deng *et al.* 2021). Once the input layer transformations have been applied to the respective images, the subsequent process of the convolution layer is applied.

2.3.2.2. Convolution Layer

Within the context of the convolution layer, hierarchical features are extracted based on the landmark points that represent the facial features as associated points (Agarap 2018; Deng *et al.* 2021; Cao *et al.* 2024). For example, complex facial structures that utilise higher level convolutional filters whilst lower levels detect simple geometric shapes (Deng *et al.* 2021; Cao *et al.* 2024). These shapes are formed through landmark points that are grouped together. In order to represent the convolution layer process, *equation 2.22* is used.

$$Y = ReLU(W * X + b) \tag{2.22}$$

The letter X is used to represent the input feature map and W is representative of the convolutional kernel. The * denotes the convolution operation where X is multiplied by W and the bias denoted by the letter b is added. The activation function defined by the ReLU symbol is then applied to the equation to ascertain the final convolution value (Agarap 2018). Upon the completion of the convolution layer process, the inception modules are applied.

2.3.2.3. Inception Modules

The purpose of the inception modules is to allow the capturing of features that vary in size. In essence it is able to inherently capture facial structural patterns based on the landmark points that are both fine-grain and broad in nature where filter sizes vary (Xu *et al.* 2019; Deng *et al.* 2021). Spatial relationships of these landmarks are therefore analysed using these inception modules that leverage parallel convolution branches (Deng *et al.* 2021). Furthermore, the parallel branches enhance the robustness of the model to variations that include those in facial pose and expression by facilitating the extraction of invariant features. The module in addition is computationally efficient since it possesses the ability to aggregate features from multiple scales whilst keeping the computational complexity at a minimum (Xu *et al.* 2019). Using a mathematical concept represented using *equation 2.23,* the concatenation of the outputs for the model's parallel branches can be represented as the inception module (Xu *et al.* 2019; Deng *et al.* 2021).

$$Y_{inception} = Concat\big(Conv_{1\times1}(X),\ Conv_{3\times3}(X), Conv_{5\times5}(X), \ldots\ldots Conv_{n\times n}(X)\big) \quad (2.23)$$

The letter *Y* is used to represent the inception module output. *X* represents the input feature map. To the input feature map, a convolution operation is applied with a filter size of $n \times n$ to receive the output feature map that is represented by $Conv_{n\times n}(X)$ (Xu *et al.* 2019; Deng *et al.* 2021). Upon the inception module operation, the residual connection process is applied.

2.3.2.4. Residual Connections

The purpose of residual connections is used to facilitate the flow of information whilst aiding gradient propagation that occurs during the training process (Sandler *et al.* 2018). This subprocess is used to establish direct connections between the input and output feature maps presented to the model. Residual connections serve to alleviate the challenge of vanishing gradients, fostering a training environment that is both stable and conducive to the efficient optimisation of deep neural networks (Lian, Wang and Zhang 2022). Through the utilisation of *equation 2.14* , highlighted in the MobileNetV2 architecture under the *residual connection* subheading, the mathematical concept is described. The next step involves the application of the global average pooling layer.

2.3.2.5. Global Average Pooling

Within neural network architectures, the global average pooling layer assumes a critical role, especially in tasks such as image classification. Its primary function involves the combination of spatial information from all feature maps (Sandler *et al.* 2018; Dan, Sun and Liu 2019; Yong *et al.* 2023). This amalgamation process serves to compress the spatial dimensions into a singular vector, capturing the core information based on the features derived from the previous layers (Üzen *et al.* 2023). Using *equation 2.15,* highlighted in the MobileNetV2 architecture under *the global average pooling layer* subheading, the mathematical concept is formally represented and described. Upon the completion of the global average pooling layer, the architecture proceeds to the next layer better known as the output layer.

2.3.2.6. Output Layer

Based on the extracted features gathered from the input data which is fed into the model, the network produces a prediction forming the output layer (Sandler *et al.* 2018). In the output layer, each neuron inherently represents a class label for classification tasks such as MFR. Within the realm of MFR, the output layer is responsible for producing the final feature embeddings based on

the extracted features (Wu and Zhang 2021; Yong *et al.* 2023). This is conducted in order to indicate the likelihood of the specified image belonging to each class within the respective categories. Through the implementation of *equation 2.17* highlighted in the MobileNetV2 architecture under the *output layer* subheading, the FaceNet InceptionResNetV1 output layer is discussed. Additionally, through this layer, compact and discriminative representations are generated, while also considering the dimensions of embeddings and considering triplet loss constraints (Cahyono, Wirawan and Rachmadi 2020). Therefore, it is important to note the pertinent subprocesses that exist within the output layer for the FaceNet InceptionResNetV1 model.

I. Dimensionality of Embeddings

The utilisation of face embedding works based on the difference and on the similarity of the successfully detected face (Wu and Zhang 2021). If the face detected, feature embedding calculations are similar to the initial face embedding calculations, the value will be closer in similarity as compared to an image where the calculations are further apart. If the calculations are further apart it symbolises that the value is further away, thus highlighting that the image may be a different person (Cahyono, Wirawan and Rachmadi 2020). FaceNet utilises an extensive dataset of face images that are used to train the model and which map faces to multidimensional feature spaces where similar faces are positioned closer together (William *et al.* 2019; Kheaksong *et al.* 2022). Hence, the balancing of the quality and computational efficiency rely on the dimensionality of the facial embeddings which need to be controlled within the output layer (Li, Deng and Chiang 2020). In mathematical terms, the dimensionality of the embeddings is reduced through the transformation of high-dimensional input X into a lower-dimensional space using a linear transformation. Using *equation 2.24,* the transformation is represented as,

$$Y = XW \tag{2.24}$$

where Y is representative of the low-dimensional embeddings. The letter W is the projection matrix that can be learned during training. The letter X is used to describe the high-dimensional input embeddings. Thus, this process allows for the capturing of essential information in a more compact and precise form (Li, Deng and Chiang 2020). The compact and discriminative representations are then discussed.

II. Compact and Discriminative Representations

A compact and discriminative representation of the input data is an aim of the output layer. A compact embedding consumes less memory and computational resources, while a discriminative embedding is able to distinguish between different classes or identities of the model (Li, Deng and Chiang 2020; Deng *et al.* 2021). To achieve this, embeddings are optimised to maximise inter-class variation while minimising intra-class variation (Li, Deng and Chiang 2020). Pertinent to the completion of a successful feature extraction process, the triplet loss is described in detail.

III. Triplet Loss

Within the FaceNet model, a negative (dissimilar to the anchor) anchor, positive (similar to the anchor) anchor and the anchor itself can be employed as three fundamental components of the triplet loss method. This is illustrated in *Figure 2. 16*.

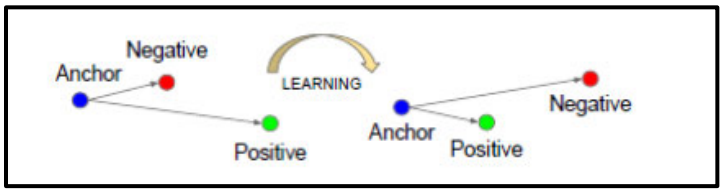


Figure 2. 16: Triplet Loss Training (Wu and Yang 2022)

The fundamental principle behind this approach involves optimising a model by minimising the distance between an anchor sample and a positive sample (from the same class) while maximising the distance between the anchor and a negative sample (from a different class). This contrastive approach helps the model learn to better distinguish between similar and dissimilar samples, enhancing its ability to classify or embed data effectively (Wu and Zhang 2021). By leveraging these three components, the triplet loss algorithm aims to optimise the feature space for effective face recognition and similarity learning (Cahyono, Wirawan and Rachmadi 2020). With the FaceNet model, an input image can be mapped to a dimensional space (D). Through the employment of the triplet-based loss method dependent on the Large Margin Nearest Neighbour (LMNN), FaceNet has the ability to efficiently train the output in order to produce a concise 128-dimensional embedding model (William *et al.* 2019; Cahyono, Wirawan and Rachmadi 2020). The development entails two sets of thumbnails, namely one for comparing faces and the other for comparing the non-

matching faces identified. The fundamental objective of the loss function is to understand and differentiate between both the positive and negative pairs within a specified range. Therefore, the thumbnails are precisely cropped. They are cropped in order to focus on the facial region. Thus, requiring no additional two or three-dimensional adjustments, except for the application of ratio and translation modifications that may occur (William *et al.* 2019; Cahyono, Wirawan and Rachmadi 2020). Using *equation 2.25*, the function is described (Wu and Yang 2022).

$$L_{triplet} = max(0, ||f(anchor) - f(positive)||^2 - || f(anchor) - f(negative) ||^2 + \alpha \quad (2.25)$$

The α represents the margin parameter, where the loss function requires positive samples and anchor embeddings to be close in distance than the negative samples and anchor embeddings. The symbol ||.|| represents the Euclidean distance. The symbol $f(.)$ is defined by the embedding function (Wu and Yang 2022).

Within part of the feature extraction process the Euclidean distance metric is utilised to quantify the similarity between feature vectors based on input numerical data fed into the model (Kortli *et al.* 2020). The Euclidean distance quantifies the geometric distance between feature vectors used in MFR, which represent unique features of faces (Kortli *et al.* 2020). By comparing the Euclidean distance between a query face and those in the dataset, a comparison to determine how closely the query face matches the dataset facial image is conducted. Thus, the smaller the Euclidean distance, the higher the degree of similarity (Kortli *et al.* 2020; Alzu'bi *et al.* 2021). The model is able to aptly identify the dataset facial image with the most similar features to the query face by comparing these numerical extracted distances input into the metric. Pertinent captured information based on facial landmarks are encoded within feature vectors. Using these feature vectors the Euclidean distance is calculated between the two facial feature vectors that are fed into the metric based on *equation 2.26* (Kortli et al. 2020; Serengil and Ozpinar 2020).

$$D = \sqrt{\sum_{i=1}^{n}(x_{2i} - x_{1i})^2} \quad (2.26)$$

Where n is representative of feature vector dimensionality and the x_{2i}, x_{1i} denote the i^{th} components of the feature vectors for the facial input values (Kortli *et al.* 2020; Serengil and Ozpinar 2020; Alzu'bi *et al.* 2021). Irrespective of the other distance metrics that exist within the

realm of machine and deep learning, such as cosine similarity or Mahala Nobis distance, which are all viable metrics, the Euclidean distance metric has a high level of popularity due to its simplicity and interpretability (Alzu'bi *et al.* 2021). Matching facial features is consistent and reliable with a metric, which is a metric space. In addition, the differences between the feature vectors are captured with both the magnitude and direction in order to provide measure of similarity that is meaningful (Serengil and Ozpinar 2020). Hence, it is a respectable choice for matching faces in this system based on the embeddings of their features (Alzu'bi *et al.* 2021).

2.4. Chapter Summary

The purpose of this chapter was to present the core theoretical frameworks, architectures and mathematical concepts required for the development of the FMD and MFR models. The chapter introduced the CNN model in context of these frameworks and architectures. Following the discussion around the CNN model, the architecture and frameworks required for the development of the FMD model was described in detail. This included a detailed description of the MobileNetV2 architecture whilst highlighting its significance in FMD. In the last section, the architecture for MFR was discussed. The MFR model comprised of several theoretical frameworks and architectures including the MTCNN, feature extraction process using FaceNet Inception ResNetV1 model and concluded with the Euclidean distance metric. The subsequent chapter presents a Systematic Literature Review (SLR) and meta-analysis, conducted in accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) protocol, providing an in-depth exploration of the various FMD and MFR models shaped by COVID-19 and other respiratory illnesses. It examines the algorithms, accuracy rates, evaluation metrics, and trends in these models, along with dataset types and limitations noted within prior research models.

CHAPTER THREE: SYSTEMATIC LITERATURE REVIEW AND META-ANALYSIS

In this chapter a detailed description based on a comprehensive exploration of Face Mask Detection (FMD) and Masked Facial Recognition models (MFR) are presented. The purpose of this chapter is to highlight the gaps identified in the existing body of research to present the need for this study. In order to highlight the significant need for this study, a Systematic Literature Review (SLR) based on the Preferred Reporting Items for Systematic Reviews and Meta Analysis (PRISMA) protocol is performed. The chapter presents the key findings from a diverse range of 72 different studies that have been conducted, providing a comprehensive overview of this critical area's evolution and current state. Furthermore, this chapter highlights the FMD and MFR meta-analysis key statistical research findings. The meta-analysis is therefore focused on the primary studies with general characteristics that are driven by the FMD and MFR development approaches, methods, dataset variations, number of images utilised, and model performances based off of the evaluation metrics. These metrics include recall or sensitivity, precision, and the f1-score or f1-measure. In the subsequent sections, the meta-analytical findings are uncovered and explored based on visual representations to highlight the trends in the FMD and MFR approaches, methods, model types, number and type of datasets utilised.

3.1. Introduction

Due to the rise of the COVID-19 pandemic as well as other respiratory diseases, it has become compulsory for individuals to wear a face mask to protect themselves and others around them from contracting this deadly disease (Wojcik and Austin 2020). Therefore, a new contactless method for verifying a user's identity without requiring physical contact with a device or system is required. It often involves biometric recognition to securely authenticate individuals, enabling convenient and fast access to sensitive information or physical property (Sarkar and Singh 2020). Facial recognition for detecting users whilst they have their face masks on has become an imperative problem that needs to be solved to allow for the effective functioning of a biometric facial recognition system. While initially motivated by the urgency in response to the difficulties presented by the global health pandemic caused by COVID-19, the research on FMD and MFR technology goes beyond its immediate application.

The progress made in this field opens up vast possibilities for diverse future use cases and scenarios (Desai and Mehrotra 2020; Wojcik and Austin 2020). The knowledge gained from tackling the COVID-19 context can be leveraged to build robust and precise systems capable of identifying individuals with face masks on. A new robust FMD and MFR model can play a crucial role in contact tracing efforts, fortifying security measures, and ensuring seamless identification processes. By embracing the lessons learned during the COVID-19 crisis, a way can be paved for transformative applications of FMD and MFR technology in a variety of contexts. Amidst the global health COVID-19 pandemic, mask usage accelerated the adoption of masked facial authentication systems, as traditional facial recognition systems became ineffective. These advanced models were designed to provide secure and contactless user authentication whilst catering to the new hygiene and safety protocols mandated by government while maintaining biometric security (Das, Ansari and Basak 2020; Mundial *et al.* 2020). The literature therefore demonstrates the evidence of several models designed and developed for FMD and MFR which is displayed in the meta-analysis below.

Annexure A provides an illustrative overview of the 72 studies that have been identified and chosen for the meta-analysis. These 72 studies elaborate on the existing literature for FMD and MFR models and further emphasise their limitations.

This chapter therefore aims to achieve the following objectives:

i. Highlight the overarching patterns within the current literature that is related to the field of FMD and MFR models, considerably in the field of deep learning, machine learning or a hybrid approach that combines both the artificial intelligence approaches.
ii. Investigate the origins of heterogeneity within the analysed studies concerning the FMD and MFR models output f1-score levels.
iii. Conduct a thorough investigation based on the potential publication bias within the analysed studies focusing on the FMD and MFR models output f1-score levels.

The developed meta-analysis is conducted using the PRISMA protocol developed by Liberati *et al.* (2009). The PRISMA protocol is utilised due to its ability to enhance inclusiveness and foster greater uniformity in systematic reviews (Sarkis-Onofre *et al.* 2021). This chapter offers a comprehensive account of the methodologies employed in conducting and documenting the meta-analysis, along with a detailed demonstration of the derived statistical outcomes.

3.2. Search Strategy

The search string below was formulated and executed on the Web of Science publication platform. By implementing the comprehensive PRISMA checklist, a meticulously crafted search strategy was developed. A thorough and expansive exploration was conducted, using the robust capabilities of the Web of Science database to pinpoint research papers of potential relevance. An evaluation based on the different methods and approaches identified to find the different models developed for FMD and MFR have been extracted from published literature articles. These articles were written in English between the years of 2019, when the COVID-19 pandemic became the focal point of all research to December 2023 (Lipworth, Chan and Kuo 2020). The following search string query was applied to extract all relevant published papers from the respective database.

((Face Mask Detection OR Mask Detection*) OR (Masked Face Recognition* OR Face recognition with a mask on) OR (Face Mask Detection* OR Mask Detection* AND Masked Face Recognition* OR Face recognition with a mask on) AND (Machine Learning* OR Deep Learning*)) (Title) and Article (Document Types) and English (Languages) and 2023 or 2022 or 2021 or 2020 or 2019 (Publication Years))*

The defined search query consisted of three main parts and was separated by 'AND' and 'OR' operators to retrieve the relevant studies related to FMD/MFR using machine learning or deep learning. The first part of the search query focuses on identifying variations of FMD and MFR or a combination of both. The second part inherently focuses on finding FMD and/or MFR using machine learning and/or deep learning. The initial search yielded 900 research papers. After applying a criteria based on articles written in English between the years of 2019 to 2023, the initial pool of 900 papers underwent refinement, resulting in a final selection of 561 eligible papers for the meta-analysis. Finally based on the search criteria pertaining to the relevance of the study, a search criterion was developed. Ultimately, a total of 72 papers were considered for the meta-analysis based on the developed criterion depicted in *Table 3. 1*. Through the utilisation of the process flow diagram (Olugbara et al. 2021) depicted in *Figure 3. 1*, the systematic literature search is perceived.

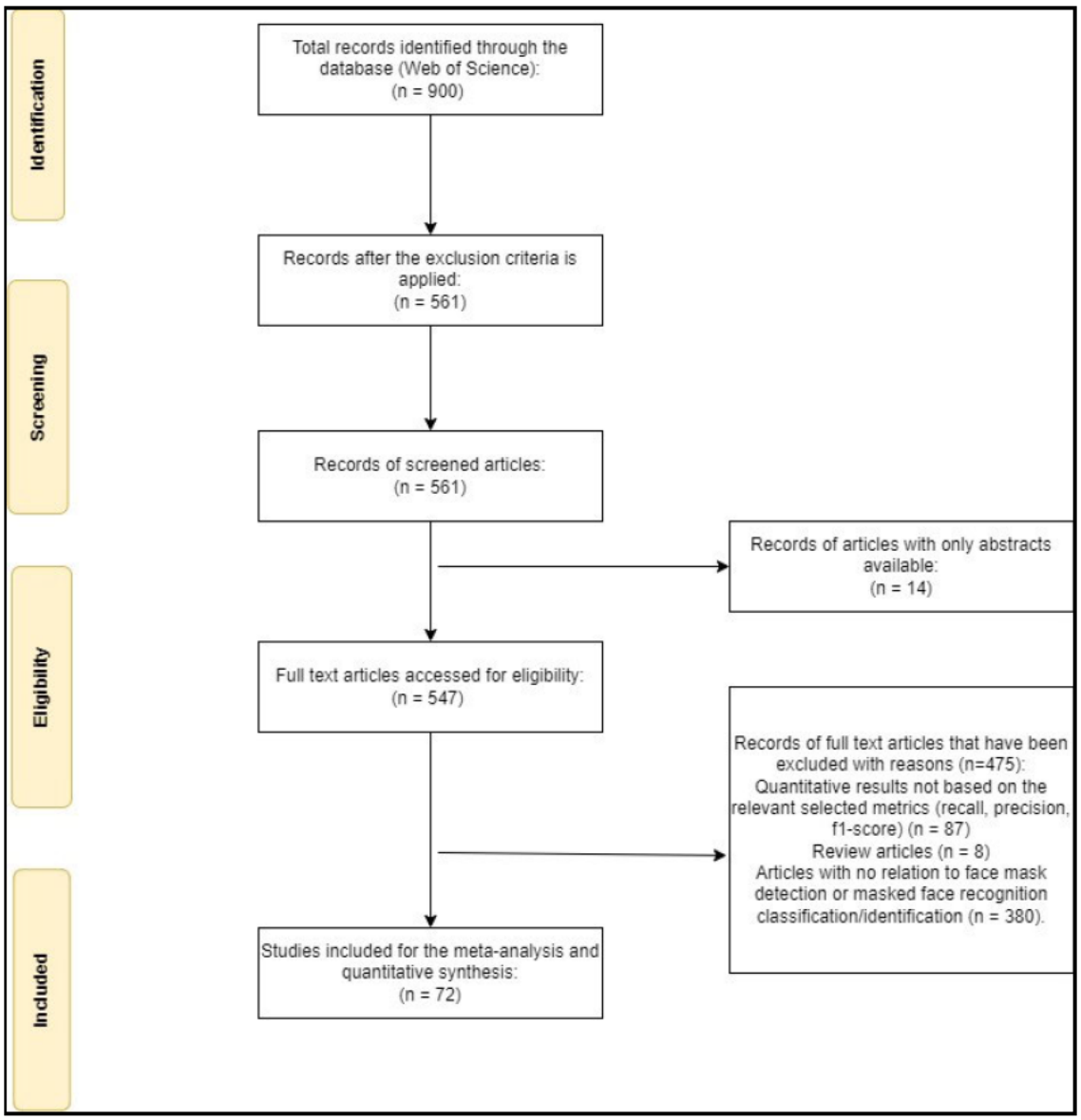


Figure 3. 1: PRISMA Protocol for FMD and MFR Models.

3.3. Selection Criteria

To ensure the most appropriate research papers were selected, an inclusion and exclusion criteria was introduced as depicted in *Table 3. 1*. The thorough automatic extraction of publications from the Web of Science database were guided by the created search query. The authors name, title of the research paper, abstract, year of publication and the journal name were extracted and outputted to a Microsoft Excel Spreadsheet for utilisation. Each publication was manually screened where the results were separated into FMD and MFR, based on the guided selection criteria highlighted in *Table 3. 1*. The final selected research papers were chosen to identify both past and current

trends in the development of FMD and MFR models that utilised machine learning and/or deep learning techniques.

Table 3. 1: An Exclusion and Inclusion Criteria

ID	Criteria
Exclusion Criteria	
EC1.	Papers that are not written in the English language.
EC2.	Papers that do not report an f1-score/measure, recall/sensitivity, and precision numeric values in the proposed study.
EC3.	Papers that do not involve a direct contribution toward the field of FMD or MFR detection, identification, and classification.
EC4.	Conference papers, survey papers, and reviews.
EC5.	Papers that do not directly relate to the proposed study of FMD, MFR or a combination of both.
EC6.	Papers that do not use deep learning or machine learning approaches.
EC7.	Papers that do not report a sample size.
ID	**Criteria**
Inclusion Criteria	
IC1.	Papers that have been published in English.
IC2.	Papers that accurately reported on the f1-score/measure, recall/sensitivity, and precision numeric values for the proposed study.
IC3.	Papers that are inherently associated with the relevant field of research (i.e., FMD and MFR).
IC4.	Papers that utilise machine learning or deep learning
IC5.	Papers that report the sample size of the proposed study.

3.4. Data Extraction

In order to extract the most pertinent information that contributes to the final meta-analysis, *Table 3. 2* provides the necessary data and information required. The general information is automatically extracted based on the relevant publication details which include the title, author, country of publication, and the publication year for each conducted study. The description of the study contains attributes such as the FMD and MFR approach, FMD and MFR models, model type, FMD and MFR methods, number of datasets, dataset names, sample information, sample sizes, accuracy, precision, recall/sensitivity, and f1-score/f1-measure. The general information is acquired through an initial screening of the papers, while details pertinent to the description of the study are meticulously collected through thorough reading and an examination of the existing

studies. After careful extraction of the desired research papers, a database was established consisting of 17 rows populated with a mixture of both quantitative and qualitative data. Therefore, the final meta-analysis consists of a matrix with 17 columns and 75 rows based on FMD and MFR models and methods. Despite the narrowed search consisting of 72 research papers, the studies were divided into FMD and MFR categories as stated in the inclusion criteria. Among the 72 papers only three research studies developed models for both FMD and MFR in combination. Hence, these three research studies were separated accordingly based on their relevant category. Therefore, the outcome displays a total of 75 papers. In the interest of providing the meta-analysis review of included studies which can be identified in *Annexure A*, only 15 columns are displayed to highlight the pertinent columns necessary for the presentation of chapter three.

Table 3. 2: Created Database Fields for Extracting Meta-Analysis Information

#	**Extraction Elements**	**Contents**	
General Information			
1	Title	Title of the article.	Text
2	Author	Credited authors of the article.	Text
3	Study Label	Quantifying a study by assigning it a numerical value for the purpose of counting.	Text
4	Country	Country of publication.	Text
5	Year	Year of publication.	Numeric
Description of the Study			
6	FMD and MFR Approach	☐ Supervised ☐ Semi-supervised	Classes
7	FMD and MFR Models	Variation of models ranging from machine learning to deep learning.	
8	Model Type	☐ Face Mask Detection(fmd) ☐ Masked Facial Recognition(mfr)	Classes
9	FMD and MFR Method	☐ Deep Learning ☐ Machine Learning ☐ Transfer and Deep Learning ☐ Machine Learning and Deep Learning	Classes
10	Number of Datasets	The quantity of datasets employed for the evaluation of the study.	Numeric
11	Dataset Names	A list of datasets used for the evaluation of the studies.	Text
12	Sample Information	The number of images utilised for each dataset.	Text
13	Sample Size	The number of data/images utilised for evaluation.	Numeric
14	Accuracy	The average accuracy for the resulted studies.	Numeric

15	Precision	The average precision for the resulted studies.	Numeric
16	Recall/Sensitivity	The average recall for the resulted studies.	Numeric
17	F1-Score/F1-Measure	The average f1-score for the resulted studies.	Numeric

3.5. Data Synthesis

Conducting data synthesis stands as an essential undertaking necessary for executing a statistical examination based on the extracted metadata (Sarkis-Onofre *et al.* 2021). The analysis involves utilising a set of meta tools, including a meta set, meta summarisation, meta-Galbraith, meta forest plot, meta funnel plot, and meta regress, all within the framework of the *Stata* software, version 18 (StataCorp 2023). This framework is a versatile and programmable statistical software package. The *Stata* software framework offers a collection of commands tailored for a comprehensive meta-analysis. This software is commonly used among researchers for conducting a meta-analysis (van den Berg, Lansu and Cillessen 2020; Borenstein 2022; Hodges *et al.* 2023). In this investigation, an examination of FMD and MFR models were undertaken, utilising meta-analytical techniques in order to calculate the impact of each individual study as well as the aggregated effect size derived from the entire set of included studies.

3.5.1. Computing Effect Sizes

In this study, the computation of effect sizes for each primary study and the collective effect size utilising gathered data on FMD and MFR is conducted. Employing the random-effects model in the meta-analysis facilitated a broader conclusion, accommodating the diverse characteristics present across the sampled studies (Hedges and Vevea, 1998). Given the array of segmentation approaches and methods observed in published articles across various image sets, the adoption of the random-effects model was considered fitting. A graphical illustration based on the effect sizes were achieved through the utilisation of a forest plot diagram, illustrating the distribution of individual study effects around zero and the pooled effect size (Olugbara et al., 2021). Additionally, the Galbraith plot was utilised to graphically present the meta-analysis results, offering insights into study-specific effect sizes, and assessing heterogeneity among them (Galbraith, 1990).

3.5.2. Detecting Statistical Heterogeneity

Across the diverse research studies included in this analysis, heterogeneity refers to differences in the results of the studies. It provides the observed differences in results beyond what could be expected by chance and are indicative of the degree to which they are outside the norm (Borenstein *et al.* 2019). It is possible to discern whether effect sizes are consistent or diverse by measuring heterogeneity using measures like I^2 and Tau^2. Managing heterogeneity is essential for precision in interpreting meta-analysis findings and may entail employing subgroup analyses or incorporating the random-effects model to accommodate diverse study variations (Borenstein *et al.* 2019; Mohan and Adler 2019).

3.5.3. Conducting Moderator Analysis

Uncovering the origins of heterogeneity is facilitated through moderator analysis, a commonly employed method aimed at testing factors influencing statistical heterogeneity in research and resolving conflicting findings in the literature (Smithson, 2018). While statistical heterogeneity metrics offer a global measure of variation, they often fail to pinpoint specific sources of heterogeneity. Consequently, the imperative for moderator analysis arises to describe the underlying causes of heterogeneity and provide a deeper understanding of the research within its context. In order to perform the moderator analysis, the subgroup analysis and the meta regression is utilised (Borenstein 2022). To facilitate the accurate evaluation of how the subsets of data compare to each other, the subgroup analysis is utilised in which data is divided into smaller groups to form the subgroup analysis. Within the context of this research, the subgroup analysis is executed. The subgroup therefore consist of the model type based on specific categories which include, FMD and MFR, and the FMD and MFR method that consist of categories including deep learning, machine learning, transfer and deep learning, and machine and deep learning. This approach aimed to delve into the diversity among the studies and explore potential heterogeneity in their findings (Smith et al., 2020). Single variable regression analysis is conducted on variables such as publication year, number of datasets, and number of images to explore potential sources of inter-study heterogeneity. Meta-regression is then employed to investigate the underlying factors contributing to this heterogeneity.

3.5.4. Examining Publication Bias

The phenomenon of publication bias occurs when the results of research are used to influence the decision to publish a study, which favours those whose results are positive (Andrews and Kasy 2019). Inherently, this type of bias results in an incomplete representation of research evidence, since inconclusive or negative findings in the respective research field is less likely to be published to read. Thus, systematic reviews and the meta-analysis may be inherently unreliable if these factors distort the overall understanding of a research topic (Van Aert, Wicherts and Van Assen 2019). In this research study, using the funnel plot (Godavitarne *et al.* 2018), publication bias can be visually examined (Marks-Anglin and Chen 2020). It is important to note that a visual representation of publication bias using a funnel plot is very subjective in nature. Through the adoption of a statistical method such as the Egger linear statistical regression test (Crocetti 2016), any underlying publication bias can be determined. Hence, a second test is conducted using statistical facts to determine the prevalence of any publication bias within this research.

3.6. Summarisation of Results

After conducting a meta-analysis based on the random effects model, the results of the 75 studies (i.e., 72 studies inclusive of the three separated previously combined studies) have been synthesised to form a quantitative set of results. This set of quantitative results are used to uncover the publication biases, heterogeneity of a statistical nature and the computed effect sizes. The meta-analysis is therefore built on the basis of the following evaluation metrics, namely the f1-score which can be calculated based on the precision and recall as well as the standard error of the effect size to estimate the performance of the model and method types identified.

3.6.1. Meta-Analysis Study Summary

In this meta-analysis summary depicted in *Table 3. 3,* using a random-effects model (DerSimonian-Laird method) with 75 studies, the overall effect size (Theta) is 2.231 with a 95% confidence interval of [0.839, 3.624]. The heterogeneity measures include Tau^2 of 7.6332, I^2 of 21.18%, and H^2 of 1.27. Individual study effect sizes, confidence intervals, and weights are presented, with the test of Theta indicating a z-value of 3.14. The test of homogeneity (Q) yielded a chi-squared value of 93.88 with a p-value of 0.0017, suggesting notable heterogeneity among the studies. The p-value for the test of Theta = 0 is less than 0.05, indicating a significant overall effect.

Table 3. 3: Meta-Analysis Study Summary

Meta-Analysis Summary Random-Effects Model, Metod: DerSimonian-Laird				
			Number of Studies: 75 **Heterogeneity:** **Tau²** = 7.6332 **I² (%)** = 21.18 **H²** = 1.27	
Study	**Effect Size**	**[95% conf.**	**Interval]**	**% weight**
Study 1 (Al-Dmour et al. 2023)	-0.158	-12.572	12.256	1.06
Study 2 (Al-Dmour et al. 2023)	-0.158	-12.572	12.256	1.06
Study 3 (Su et al. 2022)	-2.313	-14.372	9.746	1.11
Study 4 (Hu *et al.* 2022)	18.036	5.972	30.099	1.11
Study 5 (Ilyas and Ahmad 2022)	2.282	-10.357	14.921	1.03
Study 6 (Kumar 2023)	21.408	8.953	33.863	1.05
Study 7 (Ullah et al. 2022)	2.983	-8.321	14.287	1.23
Study 8 (Ullah et al. 2022)	2.983	-8.321	14.287	1.23
Study 9 (Naseri, Kurnaz and Farhan 2023)	-4.24	-15.795	6.946	1.22
Study 10 (Balasubramanian, Ramyadevi and Geetha 2023)	2.032	-10.259	14.323	1.07
Study 11 (Hussain et al. 2022)	-0.751	-13.089	11.588	1.07
Study 12 (Pham, Nguyen and Huh 2023)	12.205	3.732	20.679	1.92
Study 13 (Yu et al. 2023)	9.750	2.591	16.908	2.41
Study 14 (Bania 2023)	3.014	-9.645	15.672	1.02
Study 15 (Habib et al. 2022)	3.024	-9.636	15.684	1.02
Study 16 (Sethi, Kathuria and Kaushik 2021)	0.048	-12.039	12.135	1.10
Study 17 (Kumar et al. 2022)	-0.541	-13.192	12.109	1.02
Study 18 (Kamil et al. 2023)	3.530	-3.198	10.258	2.60
Study 19 (Loey et al. 2021)	0.572	-11.974	13.119	1.04
Study 20 (Bhaik et al. 2021)	-0.796	-11.216	9.623	1.41
Study 21 (Hussain et al. 2021)	-2.025	-14.059	10.008	1.11
Study 22 (Özyurt, Mira and Çoban 2022)	-0.541	-11.608	10.526	1.28
Study 23 (Xu et al. 2023)	0.144	-10.101	10.389	1.44
Study 24 (Kowalczyk, Sobotka and Rumiński 2023)	0.217	-9.834	10.269	1.49
Study 25 (SÜNNETCİ et al. 2023)	1.304	-11.270	13.878	1.03
Study 26 (Ottakath et al. 2022)	1.450	-9.808	12.707	1.24
Study 27 (Yahya et al. 2021)	8.251	-15.309	31.810	0.33
Study 28 (Yu and Zhang 2021)	-2.699	-15.594	10.196	0.99
Study 29 (Zhao, Zou, and Wu 2023)	-0.461	-11.610	10.687	1.26
Study 30 (Goyal et al. 2022)	-1.023	-13.175	11.130	1.10
Study 31 (Głowacka and Rumiński 2021)	3.068	-18.869	25.004	0.38
Study 32 (Kumar et al. 2021)	16.994	6.551	27.436	1.40
Study 33 (Sharma, Gautam, and Singh 2023)	19.034	7.463	30.605	1.19
Study 34 (Umer et al. 2023)	-2.114	-14.216	9.988	1.10
Study 35 (Chen et al. 2023)	-8.481	-17.063	0.101	1.88
Study 36 (Kumar, Kalia and Kalia 2022)	19.782	7.220	32.344	1.04
Study 37 (Rafidison et al. 2023)	-1.974	-10.925	6.976	1.77
Study 38 (Kumar et al. 2021)	16.994	6.551	27.436	1.40

Study 39 (Jayaswal and Dixit 2023)	-1.574	-13.415	10.267	1.14
Study 40 (Farman et al. 2022)	-1.023	-13.175	11.130	1.10
Study 41 (Dewi and Chen 2022)	-1.447	-14.427	11.553	0.98
Study 42 (Kumar and Bansal 2023)	1.654	-10.645	13.954	1.07
Study 43 (Balasubaramanian et al. 2023)	-0.563	-10.591	9.465	1.49
Study 44 (Walia et al. 2021)	11.245	-5.727	28.216	0.61
Study 45 (Nagrath et al. 2021)	-2.688	-11.947	6.572	1.68
Study 46 (Said 2020)	-0.591	-11.937	10.756	1.23
Study 47 (Teboulbi et al. 2021)	0.572	-11.974	13.119	1.04
Study 48 (Sheikh and Zafar 2023)	-1.644	-11.585	8.296	1.51
Study 49 (Benifa et al. 2023)	1.575	-11.110	14.260	1.02
Study 50 (Marwa and Kais 2022)	-0.015	-8.695	8.665	1.85
Study 51 (Wang, Li, and Zou 2023)	3.635	-8.169	15.440	1.15
Study 52 (Pann and Lee 2022)	-2.459	-15.132	10.214	1.02
Study 53 (Cimmino et al. 2022)	15.651	2.154	29.148	0.92
Study 54 (Al-Rammahi 2022)	0.572	-11.974	13.119	1.04
Study 55 (Fazeli Ardekani, Tale and Parseh 2023)	0.572	-11.974	13.119	1.04
Study 56 (Chen et al. 2022)	-1.951	-10.446	6.544	1.91
Study 57 (Mahmoud, Alharbi and Alghamdi 2022)	13.022	-11.397	37.440	0.31
Study 58 (Yu et al. 2023)	9.750	2.591	16.908	2.41
Study 59 (Peng et al. 2023)	-1.121	-12.122	9.880	1.29
Study 60 (Manzoor et al. 2022)	-0.796	-9.207	7.616	1.94
Study 61 (Talahua et al. 2021)	3.025	-9.167	15.218	1.09
Study 62 (Talahua et al. 2021)	3.025	-9.167	15.218	1.09
Study 63 (Kumar et al. 2023)	-1.571	13.665	10.522	1.10
Study 64 (Mar-Cupido et al. 2022)	0.002	-12.099	12.102	1.10
Study 65 (Hu et al. 2023)	-0.697	-6.282	4.888	3.20
Study 66 (Akingbesote et al. 2023)	-1.500	-11.538	8.539	1.49
Study 67 (Wang et al. 2023)	2.628	-3.769	9.026	2.76
Study 68 (Guo et al. 2022)	0.596	-8.302	9.494	1.79
Study 69 (Yang et al. 2022)	-2.493	-14.161	9.175	1.17
Study 70 (Zia et al. 2022)	12.663	0.602	24.725	1.11
Study 71 (Hung 2023)	-3.395	-13.305	6.514	1.52
Study 72 (Rahmani et al. 2022)	0.572	-11.974	13.119	1.04
Study 73 (Asghar et al. 2022)	-1.594	-10.449	7.261	1.80
Study 74 (Meivel *et al.* 2022)	-0.051	-5.219	5.118	3.46
Study 75 (Wang, Zhao, and Chen 2021)	-0.438	-11.273	10.396	1.32
Theta	2.231	0.839	3.624	
Test of Theta = 0: z =3.14 **Test of homogeneity = Q = chi^2(74) = 93.88**			**Prob > |z| = 0.0017** **Prob > Q = 0.0593**	

3.6.1.1. The Forest Plot

As illustrated in *Figure 3. 2*, a forest plot diagram is used to represent the weight, confidence interval, effect size and study derived from the meta-analysis. To visually display the results of multiple studies on one scale, forest plots are commonly used in the meta-analysis. The aim of conducting a forest plot diagram is to provide a comprehensive overview of individual study estimates, overall effect sizes, and the confidence intervals associated with them (Andrade 2020). An intervention or exposure can be evaluated by comparing study outcomes in the forest plot diagram, identifying heterogeneity, and identifying the overall impact. Hence, using the forest plot, complex data can be synthesised and interpreted from a diverse range of studies in a clear and concise manner since each study is represented as a line or square, and the combined effect as a diamond (Andrade 2020).

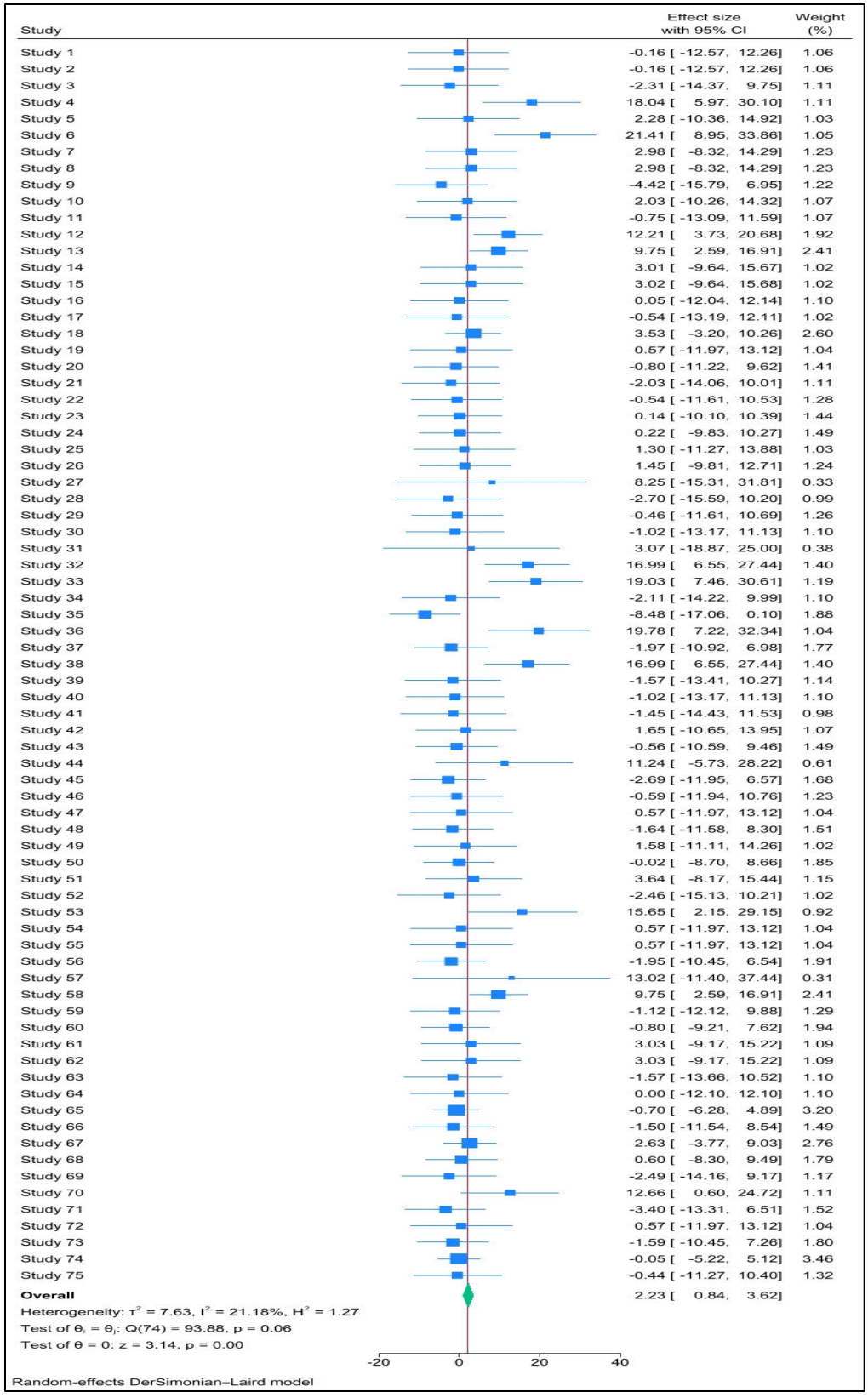


Figure 3. 2: Forest Plot Diagram

Through the implementation of the Galbraith plot as illustrated in *Figure 3. 3*, a summarisation of the meta-analysis results is conducted.

3.6.1.2. The Galbraith Plot

On the Galbraith plot, the x-axis represents the reciprocal of the standard error (z-score), while the y-axis showcases the ratio of the effect size to the standard error. In this plot each dot represents a single study of the 75 studies while the regression line cuts through the plot midpoint. As depicted in the plot below, the red line of inference which is used to represent the slope of regression is equal to 2.231, indicating the overall effect size. Studies falling beyond the shaded region of the 95% confidence interval emphasise significant diversity in the findings, indicating a notable degree of heterogeneity identified. Within the 75 studies, 66 studies (88%) fall within the grey shaded region while only nine studies (12%) in this meta-analysis fall out of the grey shaded region. This indicates a relatively low or weak heterogeneity. The Galbraith plot displays studies that exhibit greater precision as they extend further from the y-axis. In this investigation, all studies diverge from the y-axis, indicating varying levels of precision. Furthermore, heightened precision is associated with studies positioned to the right on the x-axis. Notably, nine studies extend beyond the defined region on the Galbraith plot, hinting at their potential contribution to the observed heterogeneity.

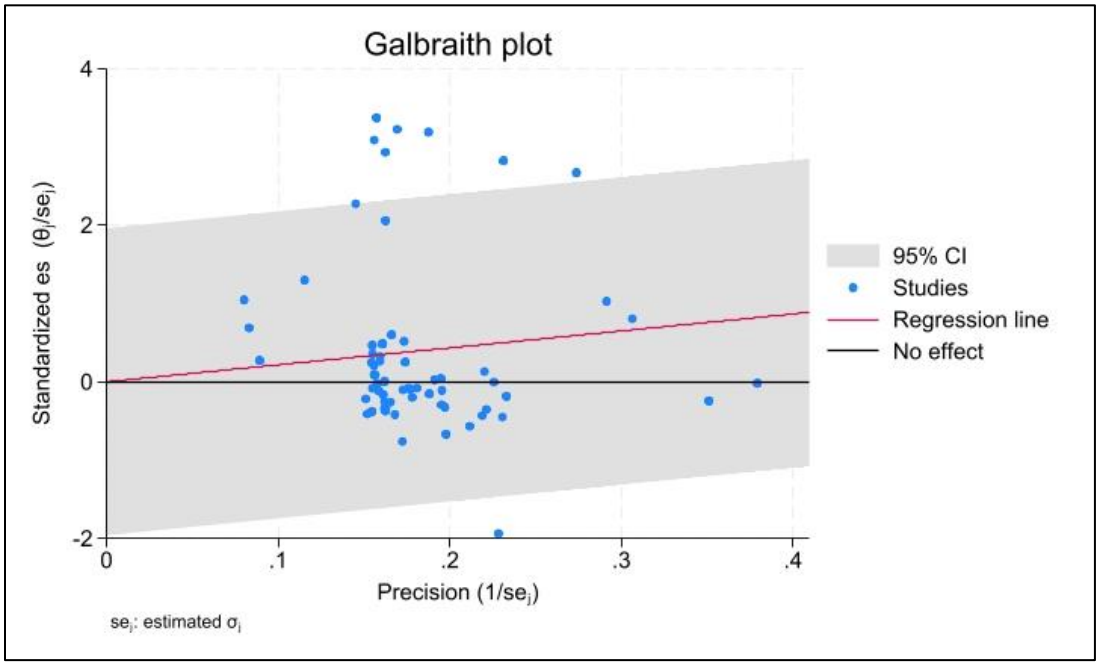


Figure 3. 3: Galbraith Plot Diagram

3.6.2. Subgroup Analysis for Different Approach Comparisons

In the context of this study, the subgroups utilised refer to the distinct categories or classifications within the dataset that were analysed separately to explore potential differences or patterns. These subgroups were defined based on specific characteristics, such as model type (*Face Mask Detection (FMD)* and *Masked Facial Recognition (MFR)*) and methodological approach (*Deep Learning, Machine Learning, Machine and Deep Learning, and Deep and Transfer Learning*). Analysing the aforementioned subgroups allow for a more refined examination of the data, thus providing insight into how different factors may influence the outcomes or effects observed in the meta-analysis.

In this subgroup analysis comparing different approaches, the meta-analysis summary as displayed in *Table 3. 4* and *Figure 3. 4*, include a total of 75 studies grouped by model type and method. For the model type, there are 64 studies in the *fmd* category with an effect size of 2.453 and a 95% confidence interval of [0.883, 4.023], yielding a significant p-value of 0.002. In the *mfr* category, there are 11 studies with an effect size of 0.839 and a confidence interval of [-2.457, 4.134], with a non-significant p-value of 0.618. Regarding the method, *Deep Learning* comprises of 60 studies with an effect size of 2.735 and a significant p-value of 0.001. *Machine Learning*, represented by two (2) studies, show an effect size of 8.120 with a confidence interval of [-3.403, 19.643] and a non-significant p-value of 0.167. *Machine Learning and Deep Learning* combined in four (4) studies have an effect size of 0.806, a confidence interval of [-4.766, 6.378], and a non-significant p-value of 0.777. *Transfer Learning and Deep Learning*, with nine (9) studies, exhibit an effect size of -1.821 and a confidence interval of [-5.349, 1.706], with a non-significant p-value of 0.312. The overall analysis, represented by theta, indicates an effect size of 2.231 with a confidence interval of [0.836, 3.624] and a significant p-value of 0.002. This subgroup analysis provides insights into the varying effects across different model types and methods in the meta-analysis.

Table 3. 4: Subgroup Analysis.

Group	Number of Studies	Effect Size	[95% conf.	Interval]	p-value
Model Type					
fmd	64	2.453	0.883	4.023	0.002
mfr	11	0.839	-2.457	4.134	0.618
Method					
Deep Learning	60	2.735	1.104	4.366	0.001
Machine Learning	2	8.120	-3.403	19.643	0.167
Machine Learning and Deep Learning	4	0.806	-4.766	6.378	0.777
Transfer Learning and Deep Learning	9	-1.821	-5.349	1.706	0.312
Overall					
Theta	75	2.231	0.836	3.624	0.002

The forest plot subgroup analysis provides a visual representation of the effect sizes and corresponding confidence intervals for different subgroups within this research study. By categorising studies based on key variables such as the FMD and MFR model type or methodological approach with deep learning, machine learning, machine and deep learning, and transfer and deep learning, the forest plot offers insights into how these factors influence the overall findings of the meta-analysis. This analysis allows for a comprehensive exploration of the heterogeneity and variability observed across subgroups, providing an enhanced understanding of the relationships under investigation within context of the research.

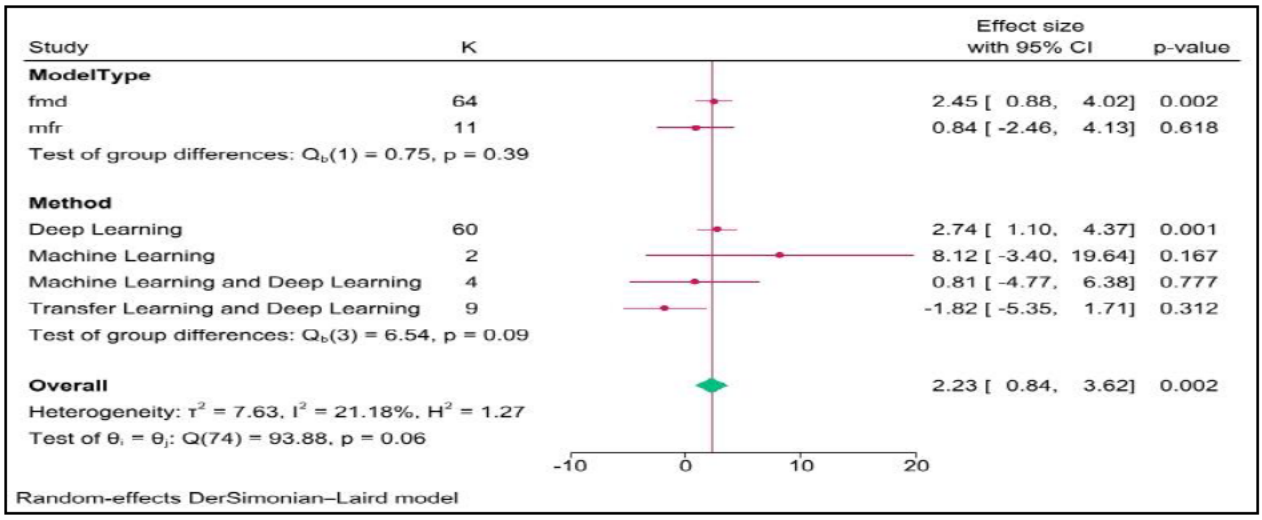


Figure 3. 4: Forest Plot Sub-Group Analysis

3.6.3. Meta Regression

The identification of statistical heterogeneity within the subgroup analysis prompts a deeper exploration of potential moderators through meta-regression analysis. Within this analytical framework, key variables including publication year, methodological techniques, dataset sizes, and image quantities are closely reviewed to uncover underlying factors driving the observed variability. This notable heterogeneity may arise from diverse sources, across variations in dataset sizes, methodological approaches, and temporal factors influencing study publication. Through a detailed examination in *Table 3. 5* and *Table 3. 6*, the meta-regression findings reveal crucial moderators shaping heterogeneity, with statistically significant revelations by providing insight based on the influence of specific variables, such as image quantities, on effect sizes.

3.6.3.1. Statistical Heterogeneity Summary

In the analysis of heterogeneity across different model types and methods as illustrated in *Table 3. 5*, notable variations are observed. For the *fmd* model type, encompassing 63 degrees of freedom, the *Q* statistic is 86.84 with a p-value of 0.025, indicating significant heterogeneity ($I^2 = 27.45\%$). Conversely, the *mfr* model type, with 10 degrees of freedom, shows a non-significant *Q* statistic (6.32, $p = 0.787$) and no observed heterogeneity ($I^2 = 0.00\%$). Among methods, *Deep Learning* exhibits significant heterogeneity ($Q = 80.79$, $p = 0.031$, $I^2 = 26.97\%$), while *Machine Learning*, represented by a single study, displays substantial heterogeneity ($Q = 2.48$, $p = 0.115$, $I^2 = 59.70\%$). *Machine Learning and Deep Learning* together, as well as *Transfer Learning and Deep Learning*, both show no significant heterogeneity ($p > 0.05$). Overall, across all studies, the *Q* statistic is 93.88 with a p-value of 0.059, indicating potential significant heterogeneity ($I^2 = 21.18\%$). This comprehensive heterogeneity summary underlines the need for careful consideration of variability in effect sizes across different model types and methods in the meta-analysis.

Table 3. 5: Statistical Heterogeneity Summary

Group	df	Q	P > Q	Tau²	% I²	H²
Model Type						
Fmd	63	86.84	0.025	10.601	27.45	1.38
Mfr	10	6.32	0.787	0.000	0.00	1.00
Method						
Deep Learning	59	80.79	0.031	10.550	26.97	1.37
Machine Learning	1	2.48	0.115	43.853	59.70	2.48
Machine Learning and Deep Learning	3	0.09	0.993	0.000	0.00	1.00
Transfer Learning and Deep Learning	8	3.40	0.907	0.000	0.00	1.00
Overall						
	74	93.88	0.059	7.632	21.18	1.27

The test of groups' differences is then used to compare the effect sizes between the different subgroups within the data.

3.6.3.2. Tests of Groups' Differences

In examining differences between model types and methods as depicted in *Table 3. 6*, the test results reveal no significant variations in effect sizes among model types (df = 1, Q_b = 0.75, p = 0.386). Similarly, for methods, the test with three (3) degrees of freedom yields a Q_b value of 6.54 with a p-value of 0.088, suggesting no statistically significant differences in effect sizes.

Table 3. 6: Tests of Groups Differences

	df	Q_b	P > Q_b
Model Type	1	0.75	0.386
Method	3	6.54	0.088

These findings therefore indicate a comparable impact across different model types and methods in the meta-analysis, supporting the consistency of results within these subgroup categories.

3.6.4. Publication Bias

As outlined in the preceding section pertaining to the examination of publication bias, *Figure 3. 5* and *Table 3. 7* are employed to provide both visual and statistical representations of this occurrence. Using the funnel plot, publication bias can be visually examined. In addition, through the adoption of the Egger linear statistical regression test, a statistical fact representation based on the underlying publication bias can be determined.

3.6.4.1. The Contour Enhanced Funnel Plot

The funnel plot is utilised to provide insight into any potential publication bias that is prevalent in the research study. As illustrated in *Figure 3. 5*, the funnel plot presents a clear asymmetrical distribution of studies which indicates a level of publication bias prevalent. The detection of studies beyond the triangular region suggests the likelihood of publication bias in the research. The vertical line signifies the combined effect size, while the diagonal lines define the 95% confidence interval in the analysis.

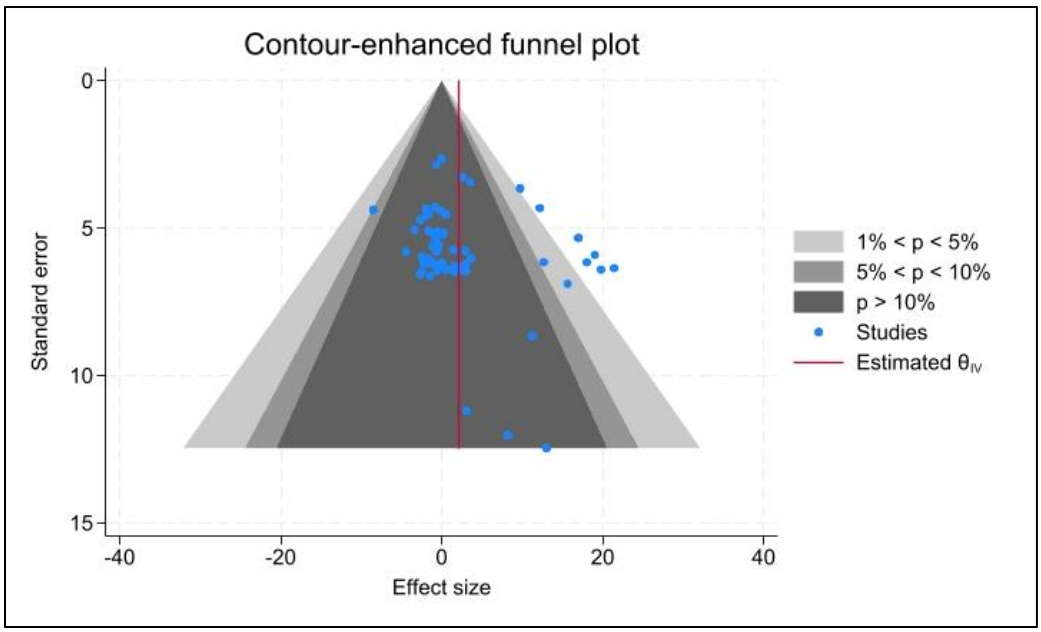


Figure 3. 5: Contour Enhanced Funnel Plot Diagram

Using Egger's test (Crocetti 2016) as depicted in *Table 3. 7,* a more accurate representation of whether there is any publication bias present using statistical analysis is presented.

3.6.4.2. Egger Linear Statistical Regression Test

The Egger test is used to assess whether there is a significant deviation from zero in the association between the effect size estimate and standard error (Page *et al.* 2021). In order to determine the statistical significance for publication bias, 0.05 was chosen for the *p* value and can be denoted as $p < 0.05$. Based on the *p*-value of 0.3297 obtained from the Egger test, it is evident that there is no form of publication bias since 0.3297 is greater than 0.05. Therefore, it is evident that the funnel plot diagram suggested evidence of initial publication bias, however after conducting Egger's quantitative statistical analysis test, it was clear and revealed non-significance, indicating the

efficacy of the study selection criteria. Hence, this indicates that asymmetry in the funnel plot may stem from factors such as sample size within the studies datasets rather than publication bias, emphasising the importance of a rigorous study inclusion criteria (Page *et al.* 2021).

Table 3. 7: Egger's Test

Regression-based Egger test for small-study effects	
Random-effects model	
Method: DerSimonian-Laird	
HΘ: beta1 = Θ; No small- study effects	
beta1 =	0.51
SE of beta1 =	0.525
Z =	0.97
Prob > |z| = 0.3297	

3.7. Discussion and Findings

After conducting a meta-analysis based on the 72 studies that met the inclusion and exclusion criteria and were published between the years of 2019 to 2023, the following outcomes were observed. The meta-analysis developed is based on identifying the current FMD and MFR models utilised within the field of machine and deep learning. Incorporating the random effects model, the meta-analysis revealed a presence of statistical heterogeneity among the effect sizes observed in the analysed studies conducted. Within context of the Galbraith plot, only 12% of the studies fell outside of the shaded region. This indicates a relatively low or weak heterogeneity. The primary determinant for the source of heterogeneity is identified as the quantity of images employed for performance evaluation. Notably, significant differences were observed in the number of images utilised for validating FMD and MFR performance across the 72 studies. The number of images utilised for each study varied from 400 in a single study to 1 000 000+ images within another study. Therefore, before drawing conclusions about the performance quality of FMD and MFR, it is essential to thoroughly examine the number of images utilised for evaluation. This underscores the significance of this aspect in ensuring a comprehensive assessment and an accurate interpretation of results are conducted (AlGerafi *et al.* 2023). Based on the conducted subgroup analysis for FMD and MFR model types and methodological approaches, the noted heterogeneity identified stems from the different sample sizes utilised through preference by each respective author for FMD and MFR model performance.

Using the funnel plot and Egger linear statistical regression test, the identification of publication bias is determined. Initially after the funnel plot was visualised and findings were noted, the funnel plot displayed a likelihood of publication bias in the research. A second test was then conducted through the application of the Egger linear statistical regression test to determine the presence of publication bias. Based on the conducted test, it is evident that there is no form of publication bias since the *p* value is equal to 0.3297 which is greater than $p > 0.05$ (Page *et al.* 2021). This finding further confirms the notion that asymmetry in the funnel plot does not solely arise from publication bias but can be attributed to other factors, such as sample size (Borenstein *et al.* 2019).

3.8. Characteristics of Primary Studies

To discern trends and uncover patterns of the 75 meta-analysis studies, *Figure 3. 6* to *Figure 3. 12* are utilised to provide further insight into the popularity of certain methods. The insights into the popularity of certain methods are conducted based on publication year, popularity of certain model types by publication year, and model type popularity by percentage value throughout all four years. In addition, it also provides insight into the approaches utilised by each study based on the number of studies. This is also illustrated in a percentage format to provide an overall depiction. Furthermore, a diagram is utilised to provide an overview based on the number of datasets used per study and to provide insight into the commonly utilised datasets per study. Lastly, a graphical map illustration is used to provide an overview and highlight the geographical based coverage of the research papers that have been published and to determine where most of the papers have been published.

3.8.1. Method Popularity by Publication Year

Figure 3. 6 highlights the trends in the reviewed publications from the years 2019 to 2023. Based on *Figure 3. 6*, it is evident that there is an increase in the development of FMD and MFR models with zero in 2019, one in 2020, 15 in 2021, and 29 in 2022. In the period of January to December 2023, a total of 30 studies were published. This marked a peak in research output for the year of 2023. Since the beginning of the COVID-19 pandemic, this deadly disease wreaked havoc causing 6 866 434 deaths worldwide according to the World Health Organisation (WHO) statistics (WHO 2023). This virus continues to spread despite the development and administration of vaccines to several people worldwide. In the year 2023, there has been at least 41 702 deaths worldwide (WHO 2023). Hence, in 2023 there was a peaked interest in the methods utilised for the development of

FMD and MFR models in order to mitigate and where possible eliminate this disease. *Figure 3. 6* reveals evidence of a pronounced trend that highlights the prevalent methods utilised which is predominantly the deep learning framework methodology. In comparison to the other methods identified, deep learning emerges as the most utilised method with the highest number of 60 studies. Transfer learning and deep learning in conjunction also have the potential to become a popular choice for FMD and MFR model development, however in 2023, deep learning having so many research studies, proves to be the most suitable method for model development.

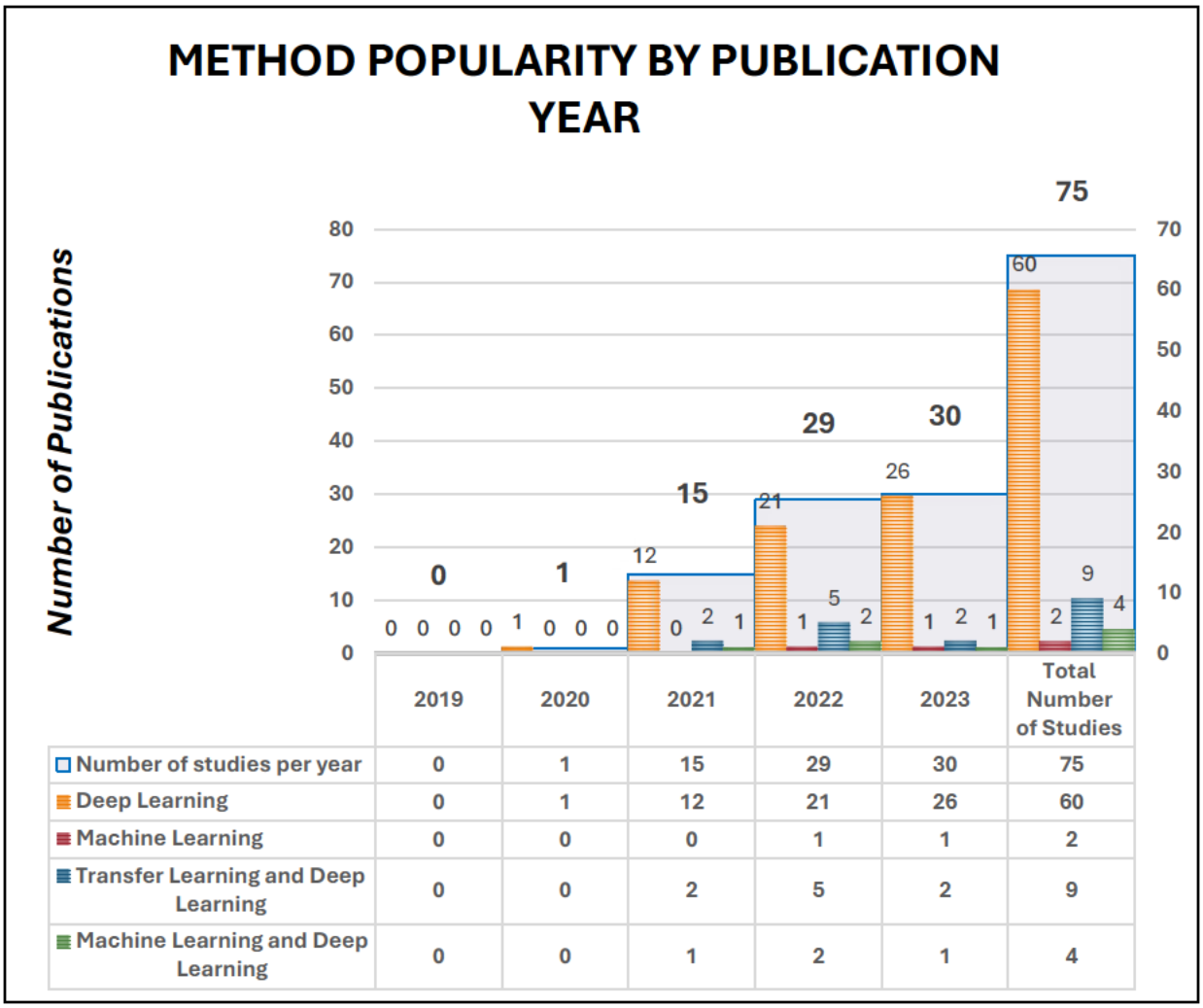


Figure 3. 6: Method Popularity by Publication Year Graph

3.8.2. Model Type Popularity by Publication Year

Within the model type popularity by publication year, the model types can be compartmentalised into two categories namely, FMD and MFR. *Figure 3. 7* illustrates how these research studies have

utilised the different model type methods throughout the years of 2019 to 2023. It is evident that there is an increase in the development of FMD and MFR models with zero in 2019, one in 2020, 15 in 2021, and 29 in 2022. In the period of January to December 2023, a total of 30 studies were published. This marked a peak in research output for the year of 2023. There is a significant number of research studies that have been produced for FMD with 64 available studies and a total of only 11 studies for MFR. *Figure 3. 8* further illustrates that 85% of the research papers have developed models for FMD and only 15% of the research papers have contributed toward the development of MFR models. The surprisingly low number of studies developed for MFR present the need for a research paper that can provide both a FMD and MFR model in combination.

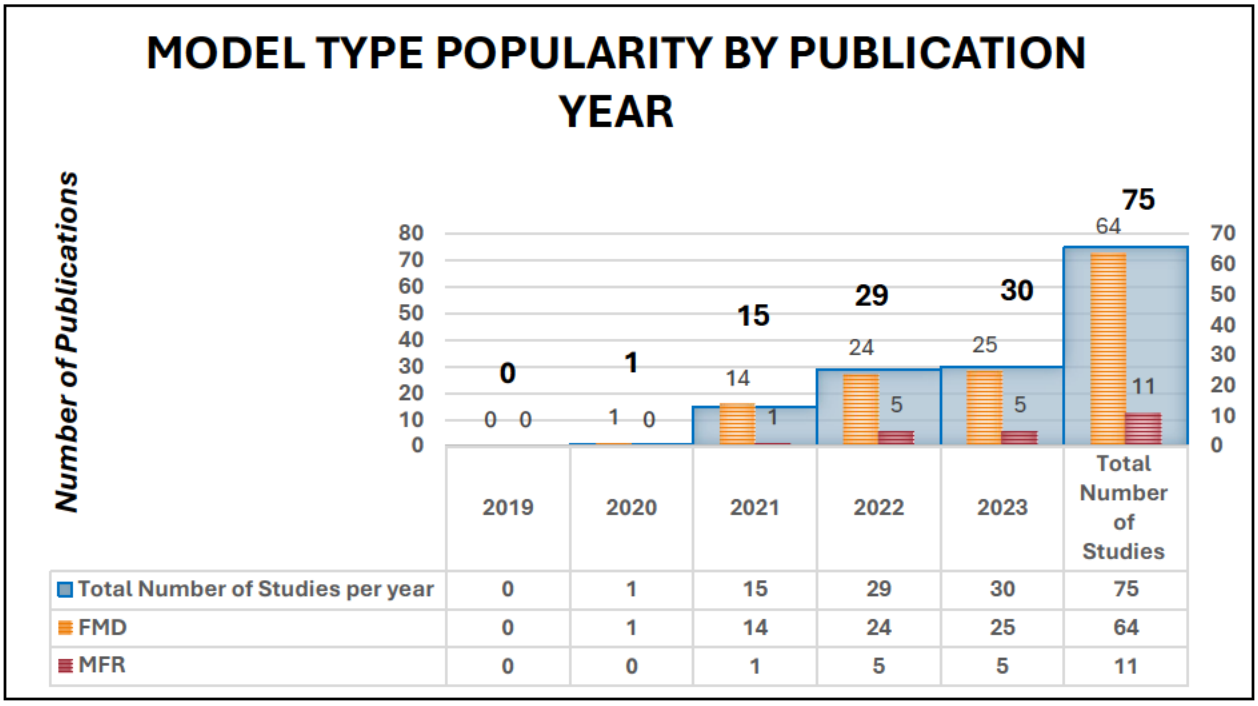


Figure 3. 7: Model Type Popularity by Publication Year Graph

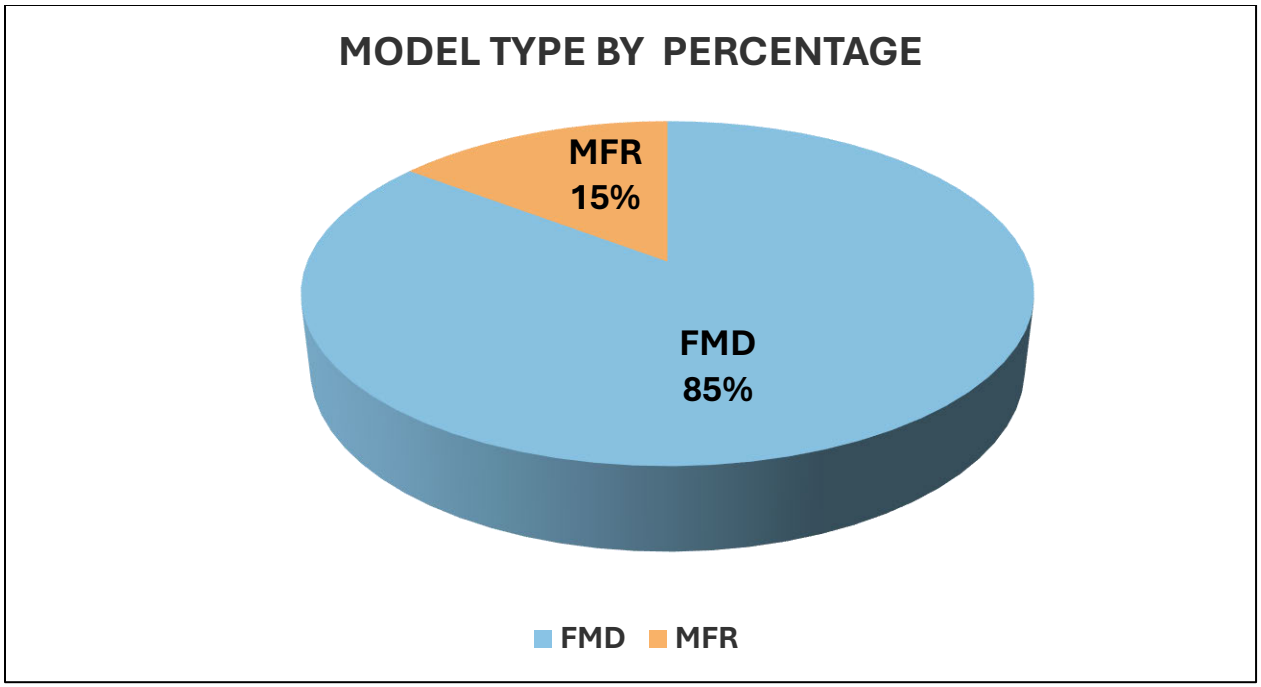


Figure 3. 8: Model Type Popularity by Percentage Graph

3.8.3. Approach by Number of Studies

Depicted in *Figure 3. 9* and *Figure 3. 10*, out of 75 research papers that were identified for the meta-analysis, 74 research papers (99%) utilised the supervised approach to develop their models. Within the 75 papers, only one research paper (1%) presented a semi-supervised approach to develop the respective research model. Given the significantly high number of research papers that were published and utilised the supervised approach to develop the FMD or MFR model, this research study utilises a supervised approach to develop the FMD and MFR model.

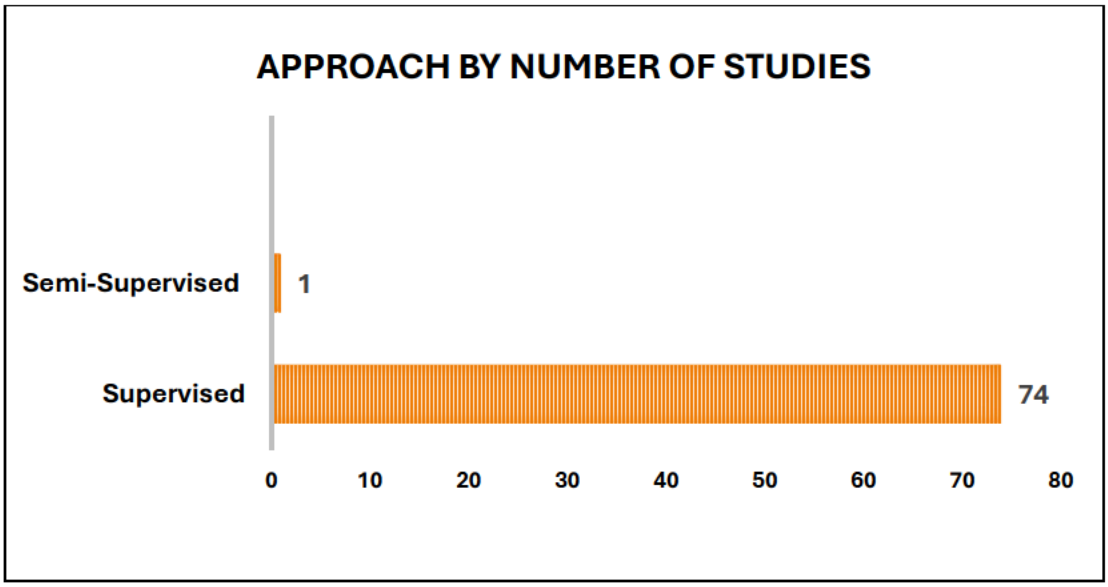


Figure 3. 9: Approach by Number of Studies Graph

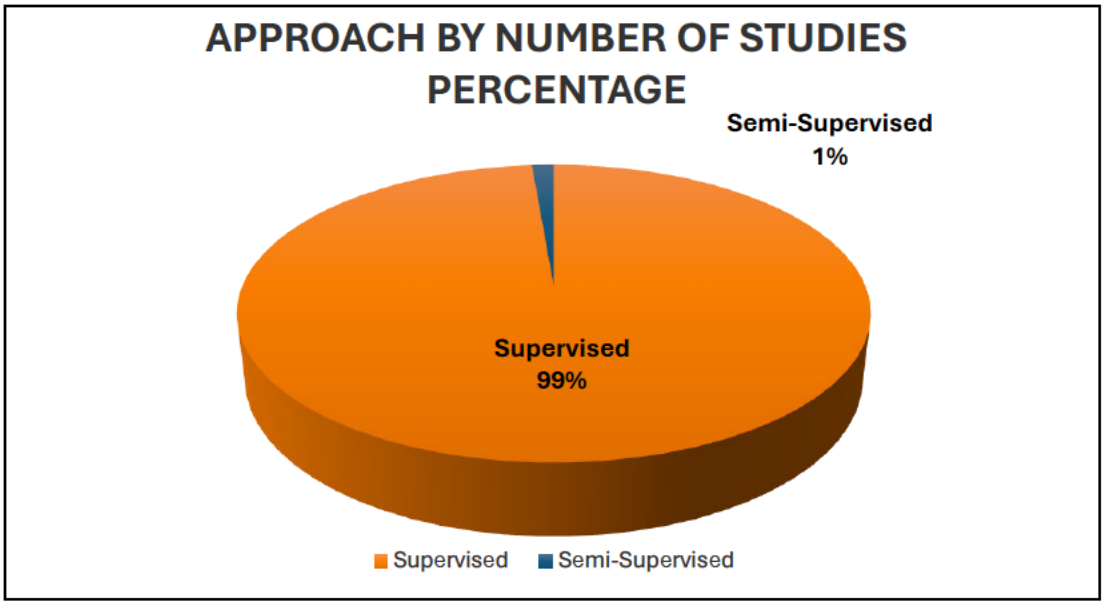


Figure 3. 10: Approach by Number of Studies Percentage Value Graph

3.8.4. Datasets by Number of Studies

It is evident that there is level of diversity in the choice of datasets and methods employed in the field of FMD and MFR. Numerous image datasets were employed in the reviewed studies to assess FMD and MFR model performance. *Figure 3. 11* presents a host of datasets that were used in the 75 research studies. A total of 54 distinct datasets were employed to train, test, and assess model performance. Within the 54 datasets, many of the reviewed studies used multiple datasets to review the performance of the models, thus creating a total of 154 datasets. The datasets are colour coded according to the number of times a dataset was used throughout the 75 studies. Throughout the meta-analysis investigation, it was prevalent that a significant number of studies utilised variations of a custom dataset that was developed for the purpose of testing the respective research models. A total of 46 studies utilised some version of a custom dataset that was created specifically for each research study. A total of eight studies utilised the RMFD dataset, and seven studies used the MAFA dataset. The Witkowski's Medical Mask, PyImage Search Reader by Prajna Bhandary, MaskedFaceNet, and Face Mask Detection Dataset by Larxel were each employed five times in the respective studies. In addition, datasets such as the AIZOO, FMD dataset by Omkar Gurav, LFW and Wider Face were utilised four times each in the respective studies. The remainder of the studies utilised a total of three or less datasets each that were employed for a performance assessment of the models. In total 47 out of the 75 studies utilised 10 000+ image that were required to train the model. This amalgamation of datasets was aimed at enhancing the robustness and generalisation of the existing literature models, a strategy that underscores the variety of approaches undertaken in the field.

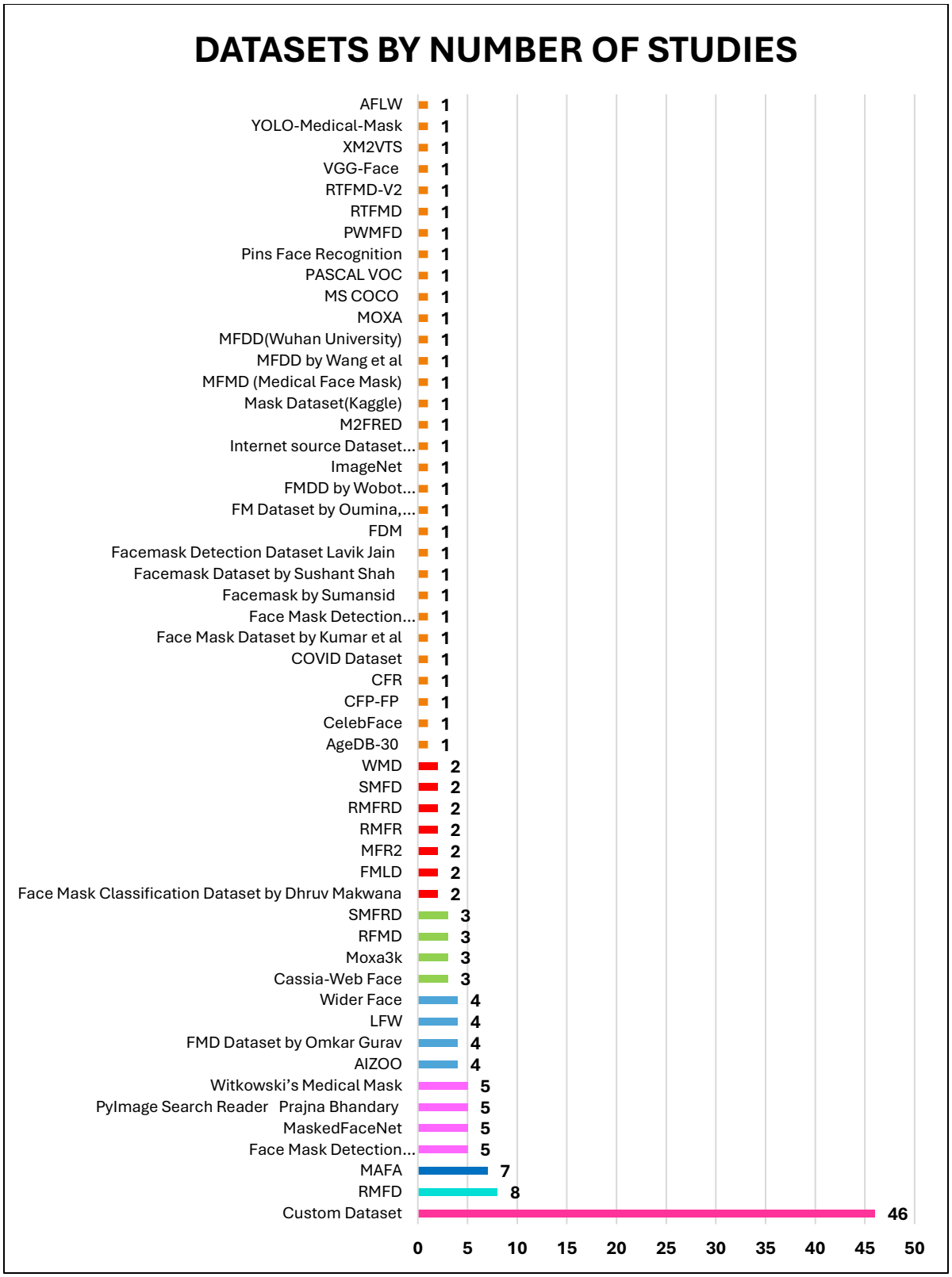


Figure 3. 11: Datasets by Number of Studies Graph

3.8.5. Spatial Distribution of Research Papers

The graphical map illustrated in *Figure 3. 12,* is used to highlight the geographical based coverage of the research papers that have been published. Furthermore, the red dot serves to emphasise the prominence of countries with a substantial volume of published research papers. Countries that contain three or more publications include, India (19), China (14), Pakistan (5), Saudi Arabia (5), Tunisia (3), Malaysia (3), and Turkey (3). Based on the graph and research findings it is evident that there were no papers published within the region of South Africa. Therefore, the absence of published research papers based on FMD and MFR using CNN in South Africa emphasises a clear research gap. Introducing a research study on this topic would provide valuable insights and contribute to addressing a pertinent need in the region as well as providing an improved study internationally.

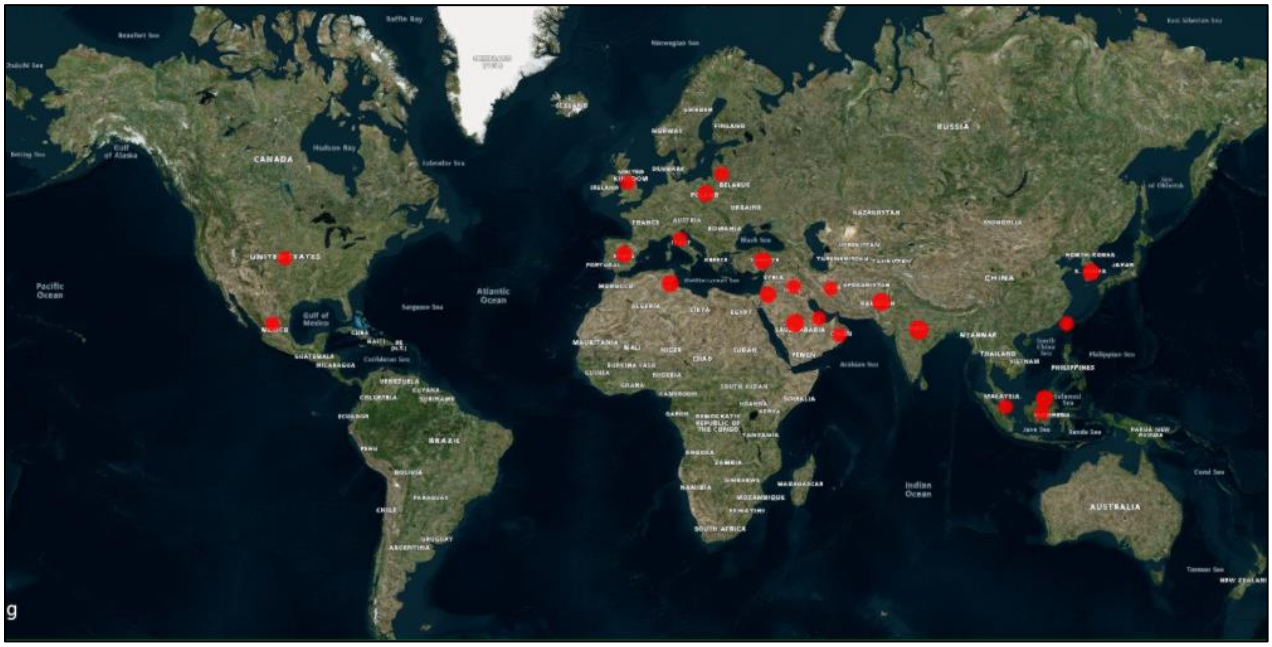


Figure 3. 12: Spatial Distribution of Research Papers

3.9. Conclusion

In recent years, the CNN has displayed immense success in computer vision and image processing tasks aptly (Khan *et al.* 2020). In demonstrating their robustness in visual data machine and deep learning applications, there has been a success rate in the development of FMD and MFR models as identified in several research papers (Islam *et al.* 2020; Mundial *et al.* 2020; Dharanesh and Rattani 2021; Talahua *et al.* 2021; Ullah *et al.* 2022). While certain studies have reported instances

of success, it is essential to acknowledge that a number of them have revealed shortcomings in delivering fully developed FMD, MFR, or a combination of the respective models. The following conclusive points encapsulate the limitations derived from the outcomes of current traditional models employed in the context of FMD and MFR.

I. The Local Binary Pattern Histogram (LBPH) is often sensitive to lighting conditions making the model less robust in different environments that the model would be presented with (Khurana, Chauhan and Singh 2020). In addition, the LBPH is also not suitable for variations in facial angles making it unfit for FMD and MFR (Arya and Tiwari 2021; Suhaimin *et al.* 2021).

II. In models such as You Only Look Once (YOLOV), real-time FMD and MFR applications are difficult to implement due to computational complexity and resource limited devices. In addition, YOLOV is not best suited for facial image recognition as it is primarily designed for object detection and presents bad training estimates and provides a generalisation for images that presents odd aspect ratios (Aswal *et al.* 2020; Mhadgut 2021). It therefore relies on numerous truth labels that are of high quality in order to train the model to receive accurate results (Zhihuan *et al.* 2018).

III. Utilisation of the Visual Geometry Group (VGG) models are not optimised for FMD and MFR since it requires extensive training to implement the model (Wei *et al.* 2020). In addition, it is often computationally expensive as it may be required to use in conjunction with other models such as CNN to implement the application (Hariri 2022) and presents a challenge when employed for real-time applications (Wei *et al.* 2020).

IV. Methods such as Eigenface demand extensive image training to precisely identify a user's face, a process acknowledged for its time-intensive nature (Kadhim, Jabber, & Hadi, 2019). In the conducted experiments, the assessments were exclusively reliant on full frontal perspectives of participants, posing challenges when users present varying facial angles to the camera, especially in instances of substantial deviations (Kadhim, Jabber, & Hadi, 2019).

V. The Fisherface algorithm, known for its higher dimensionality and increased space requirements (Reddy and Kumar, 2021), exhibits complexity in handling adverse factors such as facial expressions and illumination issues during testing (Jayaswal and Dixit,

2020). To achieve accurate results, this algorithm mandates the utilisation of an extensive number of human face images for effective training (Zarei, 2018).

The meta-analysis illuminates key findings from the literature, providing further insight on prevalent issues within context of FMD and MFR. These limitations encompass the following;

I. The literature highlights that the majority of the models focus mainly on either face mask detection (Aswal *et al.* 2020; Islam *et al.* 2020; Yang *et al.* 2020; Abbasi, Abdi and Ahmadi 2021; Suhaimin *et al.* 2021; Hariri 2022) or masked facial recognition (Mundial *et al.* 2020; Wang *et al.* 2020; Damer *et al.* 2021; Hariri 2022) and do not cater for a combination of both. A study by Kamil *et al.* (2023) in the meta-analysis highlighted that the model was only capable of FMD and not MFR, necessitating users to remove their masks for recognition authentication, thereby compromising the effectiveness of both FMD and MFR and posing a potential risk of spreading airborne diseases.
II. Both FMD and MFR models faced challenges in accurately detecting or recognising masks, with some exhibiting difficulty in detection and recognition capabilities when presented with adverse lighting conditions, resulting in extremely poor accuracy rates or inability to detect a mask or recognise a masked individual (Islam *et al.* 2020; Arya and Tiwari 2021; Boulos 2021; Mhadgut 2021; Talahua *et al.* 2021; Hussain *et al.* 2022; Marwa and Kais 2022; Kumar and Bansal 2023).
III. When presented with a variation in angles of the human face both FMD and MFR models faced challenges in effectively detecting masks or recognising masked individuals, often exhibiting limitations in their ability to perform accurate detection and recognition tasks. Thus, resulting in extremely poor accuracy rates or inability to detect or recognise masks or masked face individuals (Islam *et al.* 2020; Alzu'bi *et al.* 2021; Boulos 2021; Yu and Zhang 2021; Cimmino *et al.* 2022; Kumar, Kalia and Kalia 2022; Pann and Lee 2022; Jayaswal and Dixit 2023).
IV. Further, the existing literature revealed that the accuracy rates of FMD and MFR models can be improved significantly since a host of models produced an accuracy rate of 95% and below (Kumar *et al.* 2021; Loey *et al.* 2021; Teboulbi *et al.* 2021; Guo *et al.* 2022; Balasubaramanian *et al.* 2023; Benifa *et al.* 2023; Fazeli Ardekani, Tale and Parseh 2023; Kwak and Kim 2023).

V. Developing an efficient face mask detection and recognition model is crucial, given the significant computational complexity and resource demands of existing models (Pann and Lee 2022; Balasubramanian, Ramyadevi and Geetha 2023; Benifa *et al.* 2023; Fazeli Ardekani, Tale and Parseh 2023; Umer *et al.* 2023). Overcoming these challenges is essential for effective performance under adverse conditions. Training times vary, ranging from 1.5 hours for 12 epochs, 2.83 hours, 3.5 hours, 10 hours, to 26 hours for model training (Yu and Zhang 2021; Kumar *et al.* 2023; Umer *et al.* 2023). Studies by (Głowacka and Rumiński 2021; Nagrath *et al.* 2021; Guo *et al.* 2022; Yang *et al.* 2022); Balasubaramanian *et al.* (2023); (Peng *et al.* 2023; Rafidison *et al.* 2023) utilised epochs that ranged from 60 to 2000 that were required to train the respective models.

VI. The FMD and MFR models struggle with detecting smaller faces and low-resolution targets, often overlooking them. The models performances are affected by occlusions and noise, impacting the overall quality. The presence of noise in images poses a challenge for accurate detection and recognition, hindering model precision and introducing false positives (Hussain *et al.* 2021; Talahua *et al.* 2021; Yu and Zhang 2021; Mahmoud, Alharbi and Alghamdi 2022; Rahmani *et al.* 2022; Rafidison *et al.* 2023).

VII. It was observed that the models encountered challenges in accurately detecting instances where a face mask was worn incorrectly. Furthermore, the models were unable to differentiate situations where a person was incorrectly wearing a face mask versus not wearing one at all. Consequently, these models have not accounted for the improper use of face masks (Yu and Zhang 2021; Cimmino *et al.* 2022; Mahmoud, Alharbi and Alghamdi 2022; Rahmani *et al.* 2022; Kumar 2023; Rafidison *et al.* 2023; Sheikh and Zafar 2023).

3.10. Chapter Summary

The purpose of this chapter was to highlight the current and existing models that have been developed for FMD and MFR. A comprehensive meta-analysis of 72 research studies were conducted using the Web of Science database, revealing a wealth of pertinent research papers. Drawing conclusions from transparent and objective evidence, the meta-analysis emphasises the imperative for the development of an innovative FMD and MFR model. The next chapter delves into the methodologies and materials employed to fulfil the research objectives, particularly in development and refinement of the FMD and MFR model.

CHAPTER FOUR: RESEARCH METHODOLOGY

In this chapter, the described methodology is focused on the structure and approach of the Face Mask Detection (FMD) and Masked Facial Recognition (MFR) models. Chapter four is therefore organised into several key sections beginning with an outline through the utilisation of process flow visualisations of the aforementioned models. Following the process flow visualisation, the seven-step Convolutional Neural Network (CNN) model development is discussed in detail in context of both the FMD and MFR model. Subsequently, this chapter describes the materials utilised in developing, training, validating, and testing of the respective models, highlighting the datasets and software libraries used within the research. This systematic approach provides a robust framework utilised for understanding the intricacies of this methodology in order to build a robust model for both FMD and MFR.

4.1. Face Mask Detection and Masked Facial Recognition Process

Within the context of FMD and MFR, the methods describe the operational flow in which the models follow during the development stage. These process flow diagrams serve as a cornerstone of the research to assist in providing an accurate and efficient layout required for the development stage. Therefore, the process flows for both models are used to depict the sequence of actions the models follow from inception till completion and offer a clear and intuitive understanding of how the models function in real-world scenarios.

4.1.1. Face Mask Detection Process

The FMD process flow visualisation encompasses a series of steps aimed at the identification for the presence of a face mask on an individual's face in real-time as illustrated in *Figure 4. 1.* The process begins with the initialisation of a web camera through a local screen window. This image then goes through a series of steps to detect the input facial image presented to the model in the mask detection frame. Upon detection in this frame, the model encounters two of the following outcomes, the presence of a masked face or non-masked face. In context of a masked face, the model proceeds to assess the compliance of the masked face based on a predefined set of criteria (i.e., it covers the mouth and nose region). Hence, a green bounding box opens up around the input face and a label titled *"Mask"* with the corresponding accuracy percentage (i.e., confidence score) is displayed on the screen. In the case of a non-masked face, a green bounding box appears on the screen titled *"No Mask"* with the respective accuracy percentage. This approach can help to

enforce safety protocols and public health measures during situations that require the mandatory use of a face mask.

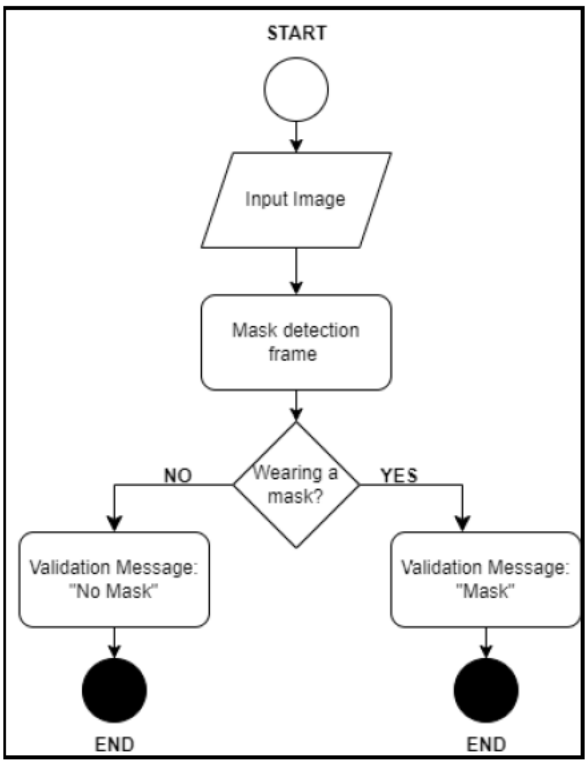


Figure 4. 1: Face Mask Detection Process Flow Diagram

4.1.2. Masked Facial Recognition Process

The MFR flow encapsulates a systematic process through a series of steps aimed at identifying and recognising masked face individuals in real-time as illustrated in *Figure 4. 2.* The process flow begins with the initialisation of a web camera through a local screen window. The input image then goes through a series of intricate steps required to recognise the input facial image presented to the model in the MFR frame. Upon detection of a masked face in this frame, the model encounters two of the following outcomes, the presence of a recognised masked face or an unknown masked face. In context of a recognised masked face, the model proceeds to evaluate the adequacy of a masked individual, assessing predefined factors such as trained and learned facial features within the MFR frame. If the correct identity of the participant is recognised, a green bounding box opens up around the participant's face. A label with the individual's name and the accuracy percentage (i.e., confidence score) of the recognised participant is displayed on the

screen. If the participant is not recognised, a red bounding box appears on the screen titled *"Unknown"* and a percentage value of "*0.00*" is set for unknown participants. This approach allows for effective and secure access control, authentication, and monitoring in high-security environments, during current and future pandemics or instances that require the mandatory use of a face mask.

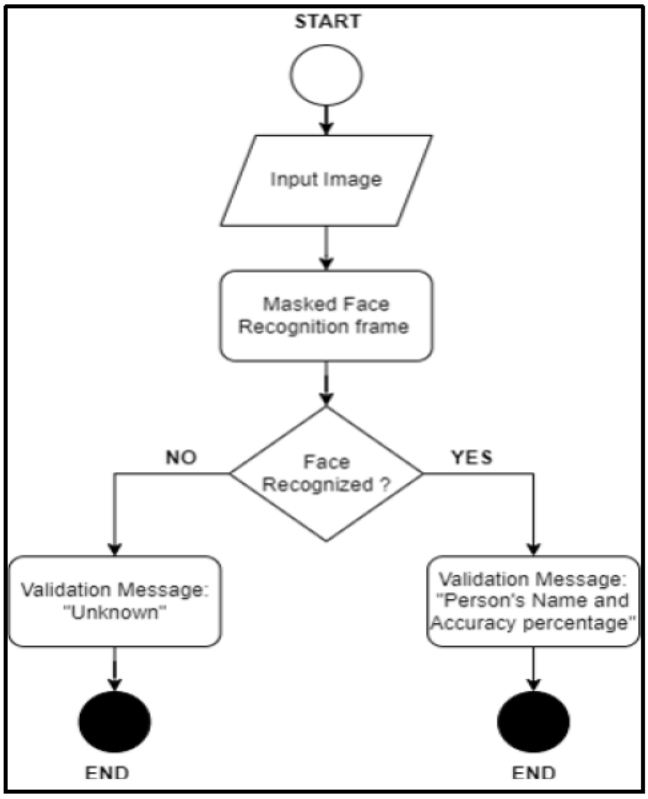


Figure 4. 2: Masked Facial Recognition Process Flow Diagram

To develop successful FMD and MFR models, it is crucial to ensure the correct algorithms and materials required to build these models are selected for development. Therefore, a detailed blueprint is a necessity in ensuring the development of these successful models. The following section consequently provides a meticulously detailed seven-step CNN model development within the context of FMD and MFR. Through a detailed description of each step, the model development design for the FMD and MFR models play a pivotal role in achieving immense accuracy in the detection and recognition of masked or non-masked individuals.

4.2. Proposed Model Development

The proposed model is developed by following the seven-step model development of the CNN architecture as illustrated in *Figure 4. 3.* Thus, the subsequent sections provide an in-depth description of each of the seven steps for both FMD and MFR model development in order to build robust models.

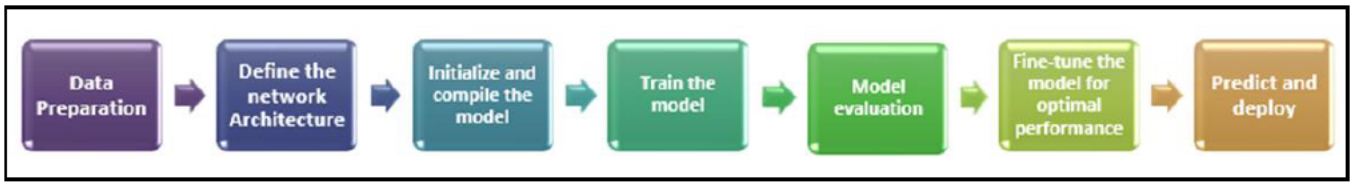


Figure 4. 3: CNN Seven-Step of the Model Development

4.2.1. Data Preparation

The data preparation step illuminates the intricacies of the data preprocessing, employed for both the FMD and MFR model development.

4.2.1.1. Face Mask Detection

Initiating the data preparation step marks a crucial preliminary phase in the development for the FMD phase of the model. The data preparation step requires a specific set of processes essential to transform the raw data into a suitable format required for training the model (Brownlee 2020). Therefore, within this step, the images undergo a series of preprocessing steps. In the first step the input images are loaded from the selected dataset. As depicted in *Figure 4. 4,* the code is developed to retrieve a specified folder from the dataset containing the images using the "data_folder". The categories list is used to define the categories or labels for the images, which in this case are *"Mask"* and *"No Mask*". The "dictionary labels", dictionary maps the numerical labels zero (0) and one (1) respectively for the *"Mask"* and *"No Mask*" categories.

```
data_folder = r'C:\Users\chezlyn\Documents\Chezlyn\Final Thesis Code\Face Mask Detection\Dataset 1'
categories = os.listdir(data_folder)
labels = [i for i in range(len(categories))]
label_in_dictionary = dict(zip(categories, labels))

data = []
target = []
img_size = 100

for category in categories:
    folder_path = os.path.join(data_folder, category)
    images = os.listdir(folder_path)
    for image in images:
        img_path = os.path.join(folder_path, image)
        img = cv2.imread(img_path)
        try:
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            resized = cv2.resize(img, (img_size, img_size))
            data.append(resized)
            target.append(label_in_dictionary[category])
        except Exception as Exp:
            print('Exception:', Exp)

# Normalisation of the image data and one-hot encode labels
data = np.array(data) / 255.0
target = np.array(target)
newer_target = np_utils.to_categorical(target)
```

Figure 4. 4: Image Preprocessing Code for FMD

Once the images have been loaded, they are converted into a greyscale format. The purpose of the greyscale image conversion is to ensure only the most critical and pertinent features are taken into consideration. In addition, this also ensures the reduction of computational complexity and the inclusion of benefiting from dimensionality reduction (Francone *et al.* 2020). Furthermore, the greyscale image format conversion maintains texture analysis and edge detection based on the original colour image among other benefits (Cascianelli *et al.* 2018; Ma and Yuan 2019). This signifies that during the conversion of a Red, Green, and Blue (RGB) image to a greyscale image format, the greyscale computational conversion process is streamlined to accommodate only for a single dimension (Francone *et al.* 2020). In an RGB image, the computation must accommodate for three dimensions which refer to the three colour channels. In a grayscale image, only a single channel requires computational processing, whereas an RGB image necessitates processing across three distinct channels, thereby increasing the computational complexity (Cascianelli *et al.* 2018; Ma and Yuan 2019). Hence, the choice of a greyscale image provides a decrease in model computational complexity.

Upon the conversion of the RGB images to a greyscale format, the images are resized and cropped to a standard scale of 100x100. Resizing the image ensures that standardisation occurs throughout all images within the dataset. Furthermore, it ensures the mitigation of small variations that occur as well as the effects of noise (Francone *et al.* 2020). Defensive programming techniques as shown in *Figure 4. 4*, are used for error handling. Resized images are normalised to pixel values between [0, 1] by dividing it by 255. The images are normalised for the purpose of ensuring there is

uniformity in data representation and to effectively facilitate convergence during the model's training (Moradi, Berangi and Minaei 2020). Upon normalisation, one-hot encoding using "np_utils.to_categorical" is then applied. One-hot encoding is crucial for converting categorical data into a numerical format, especially in classification tasks, improving model interpretation and learning (Kumar 2023).

4.2.1.2. Masked Facial Recognition

In the preprocessing step for MFR, the images are loaded and processed using OpenCV for computer vision tasks. The choice of OpenCV stems from the library's widespread adoption in image processing tasks and in its ability to process various image formats, presenting its versatility (Szeliski 2022). In addition, the OpenCV functions for computer vision are compatible with each other, allowing seamless integration into computer vision applications (Vadlapati, Velan and Varghese 2021; Szeliski 2022). The images are then converted from a BGR format to an RGB format. Within the context of OpenCV, BGR images are the default format. Hence, OpenCV defaults to loading images within a BGR format. Historically, this practice has been part of the library's development since its inception, offering historical continuity and familiarity to the image preprocessing process (Howse and Minichino 2020). Therefore, converting the image from BGR to RGB ensures seamless compatibility with the Multi-Tasked Convolutional Neural Network (MTCNN), and the FaceNet InceptionResNetV1 model by aligning the color channel order (Howse and Minichino 2020; Chunming and Ying 2021). This preprocessing step ensures that each component of the MFR model receives input in a standardised format, facilitating smooth integration and optimal performance (Cabunilas *et al.* 2023).

The model proceeds to utilise MTCNN to detect the face in the respective image. Only images that have undergone accurate detection advance to the subsequent step, wherein the image region is cropped. Once the image is detected, it is cropped and resized to 160x160 dimensions. The choice to crop the image to a size of 160x160 is driven by the prerequisites of the pre-trained FaceNet model (Sitepu *et al.* 2021). Resizing the image ensures that standardisation occurs throughout all images within the dataset. Furthermore, it ensures the mitigation of small variations that occur as well as the effects of noise (Francone *et al.* 2020). This specific dimension facilitates optimal feature extraction, a crucial step in accurate MFR. Once the image is cropped it is returned to the next step of the MTCNN function where it undergoes the landmark detection process. Hence, the

concluding stage within this function involves identification and visualisation of the facial landmark detection based on MTCNN within the *O-Net* subprocess (Francone et al. 2020; Sitepu et al. 2021). Within the landmark detection and visualisation process, in the case of an unsuccessful detection of facial landmarks based on the input image, an error message is printed, and the landmarks are excluded for that particular image. Hence, the landmark detection step is responsible for visualising and attaining the visible facial landmarks on the cropped facial image based on the information obtained from the MTCNN step.

All images that have been successfully detected, cropped and whose landmarks have been identified and visualised by the MTCNN, are further processed through a series of feature extraction techniques. These techniques include data augmentation, landmark feature extraction using the FaceNet InceptionResNetV1 model, and face matching through the utilisation of the Euclidean distance metric.

Once the localised landmark detection has been applied successfully, the next preprocessing phase focuses on feature extraction of the localised detected landmarks using the FaceNet InceptionResNetV1 model. This model is specifically trained on the 'CASIA-Webface' dataset (William *et al.* 2019). The FaceNet InceptionResNetV1 model utilises the cropped facial images and its corresponding facial landmarks as an input for the feature extraction process. The model proceeds to create copies of the input cropped facial images. It then overlays the facial landmarks on the copies and preprocesses the images for compatibility with the FaceNet InceptionResNetV1 model. The preprocessed facial images are normalised and converted into the appropriate format required for input to the model. The images are normalised to a range of approximately -1 to 1. This is achieved by subtracting 127.5 from each pixel value and then dividing by 128.0 to ensure faster convergence and improved generalisation whilst ensuring stability during the training process of the model (Moradi, Berangi and Minaei 2020). In the final step, the function utilises the pre-trained FaceNet InccpetionResNetV1 model to extract the embedded features from the processed facial images and returns these features. Once the required embedded features have been extracted using the FaceNet InccpetionResNetV1 model, the face matching process is conducted using the Euclidean distance metric.

The "match_faces" function in the developed code employs the Euclidean distance metric to calculate the distance between the facial embedding received from the preceding function and each

feature vector identified within the dataset. This distance computation is aimed at face matching. The feature vectors are flattened to streamline the calculation process, and the resulting distances are compiled into a list (Kortli *et al.* 2020). By utilising NumPy's "argmin" function, the index of the feature vector with the minimum distance is determined, effectively identifying the best match. Subsequently, the corresponding label associated with this minimum distance feature is returned by the Euclidean distance metric function. Thus, providing insight into the closest matching face within the given selected dataset (Alzu'bi *et al.* 2021).

In the preprocessing step, data augmentation is applied to the existing dataset as depicted in *Figure 4. 5.* Data augmentation stands as a widely employed technique used in computer vision tasks that exist in the machine and deep learning field (Masi *et al.* 2019). The main purpose of data augmentation is to expand and diversify an existing dataset through the utilisation of existing images within the dataset in order to reduce overfitting that may occur in models. The application of data augmentation provides a more generalised and robust model (Masi *et al.* 2019).

Figure 4. 5: Application of Data Augmentation in the MFR Model

In the data augmentation phase of this model, augmentation techniques are applied to the faces that are detected and cropped based on the MTCNN segment of the model. For each image there are 21 augmented images generated. Therefore, the new total number of images in the dataset are calculated using *equation 4.1.*

$$New\ Dataset = \big((Number\ of\ Images\ Per\ Folder\ \times\ Fixed\ Number\ of\ Augmented\ Images) + Number\ of\ Images\ Per\ Folder\big)\ \times\ Number\ of\ Folders\ in\ the\ Dataset \quad (4.1)$$

Hence, all images inclusive of the augmented images are processed through each of the relevant aforementioned detection and extraction functions. The augmentation methods involve a series of transformation techniques required to change the orientation of an image in a dataset by applying

scaling, transforming, increased noise, elastic deformations, rotations, application of lighting techniques, colour space and cropping to an image. Data augmentation techniques are applied using the "ImageDataGenerator" function of Keras. These augmentation techniques include increased rotation (20 degrees), wider horizontal shifts (0.1), heightened vertical shifts (0.2), greater shear distortion (0.3), increased zoom (0.3), and horizontal flipping. Additionally, brightness variation is introduced with a range from 0.1 to 1.0 to simulate darker conditions. Finally, channel shifting with a range of 0.2 is employed to enhance the diversity of the training data and improve the model's robustness. This can therefore be utilised during the training step of the model in order to produce better accuracy and generalisation of the model.

Using an "NPZ" file, the extracted feature numerical values are then saved for storage to be utilised within the training process. The purpose of storing the extracted feature numerical values are useful for the purpose of saving and loading multiple NumPy arrays in an efficient manner. In addition, the compression used in a npz file helps to reduce the file size. This therefore makes it easier to store and transfer a large amount of numerical data efficiently (Zhang *et al.* 2020).

4.2.2. Convolutional Neural Network Architecture Definition

The CNN architecture defined for building both the FMD and MFR models, are detailed below highlighting the defined network architecture.

4.2.2.1. Face Mask Detection

In the network architecture definition step for FMD, the model focuses on leveraging the power of the Keras sequential Application Programming Interface (API). The model incorporates the pre-trained MobileNetV2 architecture which is a variation of CNN designed for efficient image processing tasks (Sandler et al. 2018). MobileNetV2 is chosen for its lightweight structure and high performance, making it well-suited for real-time face detection applications (Rokhana, Herulambang and Indraswari 2021). The Keras Sequential API proves advantageous in constructing the model layer-by-layer, seamlessly integrating the specified layers into the framework (Gao, Zhang, and Wei, 2018). The model comprises of an input layer followed by the MobileNetV2 base model, which is loaded with pre-trained weights from the 'ImageNet' dataset. To adapt the base model to the proposed FMD model, custom classification layers are added on top of the base model. These include global average pooling, dense layers with "Rectified Linear Unit" (ReLU) activation functions, dropout layers for regularisation, and a final dense layer with

a “Sigmoid” activation function for binary classification (Toppo et al. 2022). The Sigmoid activation function outputs a probability score between zero (0) and one (1), indicating the likelihood of a person wearing a mask (0) or not wearing a mask (1). The activation function therefore accommodates the two exclusive classes for *“Mask”* or *“No Mask”* (Salehi et al. 2023). These convolutional layers are designed to capture complex patterns in both masked and unmasked faces. The ReLU activation function is applied to the neurons within these layers. ReLU promotes non-linearity in feature representation and effectively mitigates the vanishing gradient problem (Dubey, Singh, and Chaudhuri 2022). Within the hidden layers of the MobileNetV2 architecture, a default stride of 1x1 is utilised for all convolutional layers, facilitating comprehensive feature extraction by examining the input feature maps with no gaps. Following this, dropout regularisation is incorporated into the model with a dropout rate of 0.5, placed after each dense layer to mitigate overfitting during training. Subsequently, the global average pooling layer is applied to condense the extracted features into a single value per channel, effectively producing a 1x1 feature map while retaining essential information. Dense layers with varying neuron counts, specifically 512, 256, 128, and 64, are then incorporated to introduce non-linearity and further abstraction, facilitating the capture of intricate features in context of FMD. This carefully orchestrated architecture aims to optimise feature representation and model performance for the FMD model, ensuring effective FMD is conducted while improving the model's generalisation capability and robustness to unseen data.

4.2.2.2. Masked Facial Recognition

In this step, the MFR model is defined by a CNN architecture. The model comprises of three hidden layers, carefully balancing computational efficiency and feature extraction capabilities. The initial layer consists of 128 neurons, followed by a MaxPooling layer with a pool size of two for effective down-sampling. The subsequent layer features 256 neurons, enhancing the model's ability to discern masked facial attributes. Another MaxPooling layer with a pool size of two maintains the down-sampling strategy. The model then flattens the output into a one-dimensional vector. To enhance the network's capacity without overfitting, a dense layer with 512 neurons is introduced, complemented by a dropout rate of 0.5 and the addition of batch normalisation. Batch normalisation enhances and expedites the training process by standardising the inputs across each layer, resulting in improved stability and efficiency (Ioffe and Szegedy, 2015). Furthermore, the ReLU activation function aids in effectively discerning patterns in masked faces (Ullah *et al.*

2022). In the final step, the “SoftMax” activation function is applied to the output layer to accommodate multi-class classification requirements. SoftMax inherently converts the raw outputs of the model into a probability distribution across multiple classes. It ensures that the predicted class probabilities sum to one, making it easier to identify the most likely class. This is essential for determining which class an input belongs to in the MFR multi-class classification model (Grandini, Bagli and Visani 2020). This refined architecture represents a strategic enhancement of the MFR model, developed to deliver improved results in identifying masked face patterns.

4.2.3. Initialisation and Compilation of the Model

Once the network architecture has been defined using the respective network architectures, the next step in the seven-step CNN architecture is focused on the initialisation and compilation of the specified described models.

4.2.3.1. Face Mask Detection

The FMD model is initialised with the “Adam” optimiser, and a lower learning rate of 0.0001 is set. This optimiser's versatility and performance across various models make it a suitable choice since it does not require extensive parameter tuning and can handle varying gradients efficiently during training due to its adaptive learning rate (Wang, Chen and He 2020; Yi, Ahn and Ji 2020). The learning rate of 0.0001 is effectively chosen to ensure gradual and stable convergence during training. Hence, it balances between fine-tuning the pre-trained model's weights and updating the custom CNN layers effectively and efficiently (Wang, Chen, & He, 2020). In the loss function, “binary_crossentropy” is applied, enabling the model to optimise its parameters effectively by measuring the dissimilarity between predicted and actual mask presence (Toppo et al. 2022). Weights and biases are incorporated into the MobileNetV2 model during training to adjust its parameters iteratively and enhance its performance through optimisation algorithms such as gradient descent. These parameters are fine-tuned alongside custom classification layers to improve the model's ability to distinguish between mask and no-mask classes (Alzubaidi *et al.* 2021).

4.2.3.2. Masked Facial Recognition

The MFR model is also compiled using the “Adam” optimiser with a reduced initial learning rate of 0.0005 to effectively ensure gradual and stable convergence during training. The loss function is utilised using “categorical_crossentropy”. The choice to employ the "categorical_crossentropy"

function stems from the network's adaptable nature to accommodate for different scenarios, variations and can measure the dissimilarity between predicted and actual classes, for model optimisation (Maji and Gupta 2023). The weights and biases of the CNN model are initialised randomly during model creation. Upon initialisation, the weights and biases are updated during training through backpropagation. During training, the model learns optimal values for these parameters to minimise the loss function and improve performance on MFR image classification (Ayyadevara 2018; Alzubaidi *et al.* 2021).

4.2.4. Model Training

The model training step is dedicated to training the model with a primary emphasis on acquiring a comprehensive grasp of the underlying data patterns discovered.

4.2.4.1. Face Mask Detection

During the training process, the data is split into an 80/10/10 percentage. In this step, 80% of the data is utilised during the training phase of the model. The remaining 20% of the data is divided evenly, with 10% allocated for validation and 10% reserved for testing. The validation data is inherently utilised for fine-tuning the model's hyperparameters to ensure generalisation and prevent model overfitting whilst the testing data is used to determine the generalisation of the model on new unseen data (Weerts, Mueller and Vanschoren 2020; Hu *et al.* 2022). The 80/10/10 split is chosen due to its popularity amongst research papers (Chavda *et al.* 2020; Kumar *et al.* 2021; Kumar, Kalia and Kalia 2022; Kumar *et al.* 2023). In addition to its popularity, the split provides a good balance between obtaining sufficient data for training and validating the model to effectively learn the discovered patterns and generalise them, whilst still preserving a sufficient amount of data required for testing the model on unseen data. Therefore, this split assists in identifying the occurrence of overfitting within the data as well as aiding in guiding parameter tuning of the FMD model (Joseph and Vakayil 2022). A "stratify" parameter is applied in the 'train_test_split' function to ensure that the data split maintains the same class distribution as the original dataset, helping to preserve the balance of classes in the training, validation and testing sets (Hu et al. 2022). Within the training phase of the model, data augmentation techniques are applied as depicted in *Figure 4.* 6. The application of data augmentation helps to address class imbalance issues by creating augmented samples for both the classes to aid in a more balanced training set and to mitigate biases in the model's predictions (Shorten and Khoshgoftaar 2019).

The data augmentation employed the “ImageDataGenerator” function, which introduces various transformations such as rotation, shifting, shearing, and flipping to enhance the model's robustness. The data augmentation techniques applied include rotation with a range of 40 degrees, horizontal and vertical shifts with a maximum magnitude of 0.2 of the image dimensions, shear distortion with a magnitude of 0.2, zoom with a maximum magnification of 0.2, and horizontal flipping. Additionally, the "nearest" fill mode is used to handle any empty pixels created by the augmentation process. These techniques collectively aid in creating a more diverse and comprehensive dataset for training the FMD model.

Figure 4. 6: Application of Data Augmentation in the FMD Model (Jangra 2020)

Upon augmentation, a number of batch sizes were experimented on (i.e., 32, 64, 128 and 256) and a final batch size of 32 is chosen as it displayed the most effective outcome during the model’s training in comparison to the other batch sizes applied and tested on. The model is trained with augmented training data and labels using 35 epochs. The choice of 35 epochs balances the training time and model performance, allowing sufficient learning from the augmented data without overfitting. This number is inherently selected based on the early stopping criteria and performance evaluation across the aforementioned validation metrics (Bejani and Ghatee 2021). Selecting 35 epochs therefore provides an effective balance between convergence and generalisation within the model. Monitoring the training process confirmed that 35 epochs provided the best accuracy and loss results. Thus, the model is able to learn intricate patterns within the dataset as it refines its weights, resulting in improved performance without increasing training time significantly. The validation data is subsequently employed to optimise the model's hyperparameters and facilitate adjustments during the training process, thereby ensuring effective generalisation while mitigating the risk of overfitting (Joseph and Vakayil 2022). The training process is enhanced with callbacks, including a learning rate scheduler, and early stopping, promoting efficient learning, and

preventing overfitting. This comprehensive approach, utilising MobileNetV2 and incorporating custom classification layers, ensures that the model is well-equipped to detect faces with or without masks effectively. The chosen architecture and training strategies aim to strike a balance between capturing intricate mask features and preventing overfitting. In the framework of this model, a patience parameter is established, specifically implemented within the "Early Stopping" callback, which is set at a value of 10. This parameter dictates that the training process halts if there is a lack of improvement in the validation accuracy over 10 consecutive epochs. This strategic utilisation of patience is instrumental in mitigating overfitting risks, ensuring the model converges effectively, and attains optimal performance (Muhammad *et al.* 2022).

4.2.4.2. Masked Facial Recognition

Step four is aimed at training the MFR model. During the training process, the data is also split into an 80/10/10 split. In this step, 80% of the data is utilised during the training phase of the model. The remaining 20% of the data is divided evenly, with 10% allocated for validation and 10% reserved for testing. The validation data is utilised to fine-tune hyperparameters, promoting generalisation and avoiding overfitting, whilst the testing data evaluates the model's performance on unseen data (Weerts, Mueller and Vanschoren 2020; Hu *et al.* 2022). In this step, the number of epochs required to train the model is also determined. The model is trained with the training data and labels using 35 epochs. The choice of 35 epochs balances training time and performance, enabling sufficient learning without overfitting and yielding optimal accuracy and loss results (Bejani and Ghatee 2021). Therefore, it is able to learn intricate patterns within the dataset as it refines its weights, resulting in improved performance without increasing training time significantly. Validation data is then employed to optimise the model's hyperparameters and facilitate adjustments during the training process, thereby ensuring effective generalisation while mitigating the risk of overfitting (Joseph and Vakayil 2022). A number of batch sizes were experimented on (i.e., 32, 64, 128, 256 and 512) and a final batch size of 128 is chosen as it displayed the most effective outcome during the model's training in comparison to the other batch sizes that were tested on. In addition, a custom learning rate scheduler is defined to gradually decrease the learning rate after the initial epochs. The learning rate scheduler, coupled with early stopping, model checkpointing, and a dynamic reduction of the learning rate on form, contributes to the overall robustness and generalisation of the model. Within this model, a patience parameter is set. The patience parameter is set to 10 in the "Early Stopping" callback. This implies that

training inherently stops if there is no improvement in validation accuracy for 10 consecutive epochs in the initiative to overcome and prevent the occurrence of overfitting and ensuring the model converges to the best performance (Muhammad *et al.* 2022).

4.2.5. Model Evaluation

Upon the completion of the training step, evaluating the effectiveness and accuracy of the models are the next important step in the seven-step CNN model development. In this step a meticulous assessment based on both the models performance are conducted. Using the various metrics mentioned below, the models' effectiveness is essentially determined to gain insight into the performance.

4.2.5.1. Face Mask Detection and Masked Facial Recognition

Performance metrics are often applied to determine how well the model has performed (Nuruddin Qaisar Bhuiyan *et al.* 2019). Both the FMD and MFR models are evaluated using the confusion matrix to determine the accuracy of the model. The confusion matrix is one of the desired accuracy metrics selected since it analyses the performance of a classification model in a comprehensive and intuitive manner (Deng *et al.* 2016). In addition, the FMD and MFR models are also evaluated based on the widely employed evaluation metrics which include the accuracy, precision, recall, and f1-score (Grandini, Bagli and Visani 2020). Therefore, both models apply the same metrics to assess the models' effectiveness on the unseen test data. Thus, it highlights the model's significance in determining whether the proposed models performed optimally.

I. Confusion Matrix

The confusion matrix is used for diagnosing specific problems and improving the model, this information provides granular insights into the different types of errors and intuitive key indicators the model performs (Deng *et al.* 2016). The four factors in a confusion matrix (Mundial *et al.* 2020; Damer *et al.* 2021; Dharanesh and Rattani 2021; Suhaimin *et al.* 2021; Ullah *et al.* 2022) are expressed below.

- **True Positive (TP) -** This refers to the ability to correctly identify a masked face during recognition and then match it to the corresponding training dataset image (positive label) accurately. For instance, in MFR, when a person wearing a mask is correctly recognised with their face mask on or in the instance of FMD, when a person is correctly wearing their face mask, and the model

correctly associates it with the *"Mask"* class (Yi, Ahn and Ji 2020; Dharanesh and Rattani 2021; Suhaimin *et al.* 2021; Ullah *et al.* 2022).

- **True Negative (TN) -** This occurs when an unrecognised masked face (negative label) that does not belong to the dataset remains unidentified after training. For example, when a face that is not part of the dataset is labelled as *"Unknown"* or in the instance of FMD, when a person that does not have a face mask on is correctly labelled as *"No Mask"* (Yi, Ahn and Ji 2020; Dharanesh and Rattani 2021; Suhaimin *et al.* 2021; Ullah *et al.* 2022).
- **False Positive (FP) -** FP is observed when an unknown face in the classification process is wrongly identified as a participant that exists within the dataset (positive label). This can happen if an unrecognised face is mistakenly matched with a known individual wearing a mask from the dataset. In the case of FMD, a false positive occurs when the model incorrectly identifies a non-masked face as wearing a mask (Yi, Ahn and Ji 2020; Dharanesh and Rattani 2021; Ullah *et al.* 2022).
- **False Negative (FN) -** FN in MFR arises when a masked facial image observed during classification does not correctly correspond to the accurate negative label. An example of this is when a masked face from the dataset, which should be recognised, is incorrectly labelled as *"Unknown"*. In FMD, a false negative occurs when the model incorrectly identifies a face without a mask as having one on (Deeba et al. 2019; Sailusha et al. 2020).

II. Evaluation Metrics

In evaluating model performance, key metrics such as accuracy, precision, recall, and the f1-score, are essential benchmarks to accurately assess model performance (Yi, Ahn and Ji 2020; Suhaimin *et al.* 2021; Ullah *et al.* 2022). These metrics collectively provide a comprehensive assessment of the model's ability to correctly identify and classify instances when presented to it.

- **Accuracy** - The accuracy or the correctness index is computed as indicated in *equation 4.2.* This metric offers a comprehensive evaluation of the model's effectiveness by determining the ratio of accurately predicted instances to the

total number of instances. However, its suitability may be compromised in datasets with significant class imbalances.

$$Accuracy = (TPR + TNR) / (TPR + TNR + FPR + FNR) \quad (4.2)$$

- **Precision -** The precision or precision index, is expressed in *equation 4. 3*. This metric measures a model's proficiency in avoiding false positives, particularly when the consequences of false positives are substantial. This index assesses the accuracy of predicting true positives (Abbasi, Abdi and Ahmadi 2021; Dharanesh and Rattani 2021; Mandal, Okeukwu and Theis 2021; Septi, Yulita and Napitupulu 2021; Ullah *et al.* 2022).

$$Precision = TPR / (TPR + FPR) \quad (4.3)$$

- **Recall -** The recall or sensitivity score, can be calculated as depicted in *equation 4.4*. The recall metric is used to assess the proportion of true positive predictions relative to all actual positive instances. This metric gains significance, especially when the cost of false negatives is considerable (Abbasi, Abdi and Ahmadi 2021; Damer *et al.* 2021; Dharanesh and Rattani 2021; Mandal, Okeukwu and Theis 2021; Septi, Yulita and Napitupulu 2021; Ullah *et al.* 2022).

$$Recall = TPR/(TPR + FNR) \quad (4.4)$$

- **F1-score -** The f1-score serves as an evaluation metric derived from the harmonic mean of precision and recall, as expressed in *equation 4.5*. Utilising the f1-score to assess model performance provides a balanced consideration of both precision and recall, proving particularly beneficial in scenarios with uneven class distributions (Boulos 2021; Damer *et al.* 2021; Mandal, Okeukwu and Theis 2021; Ullah *et al.* 2022)

$$F1 - Score = (2 * precision * recall) / (precision + recall) \quad (4.5)$$

4.2.6. Model Fine Tuning

Fine-tuning the FMD and MFR models in step six is crucial for adjusting existing models to enhance their ability to detect face masks and recognise faces, especially when individuals are wearing face masks. This iterative process allows models to learn intricate patterns and features

associated with masked faces, enhancing their accuracy and reliability (Yi, Ahn and Ji 2020; Toppo *et al.* 2022).

4.2.6.1. Face Mask Detection

Within context of fine-tuning the pre-trained MobileNetV2 model for FMD, adjusting layers within this architecture is conducted. In the FMD model, custom layers are added on top to the base of the model to adjust it for FMD classification. These additional layers include dense layers with dropout and batch normalisation for feature extraction and classification of masked and non-masked faces (Ding *et al.* 2019). In addition, a subset of layers in the MobileNetV2 base model is unfrozen and trainable, while others remain frozen. The code unfreezes and fine-tunes a subset of layers in the MobileNetV2 base model by iterating through each layer, setting them to be trainable, except for the last 20 layers. This selective unfreezing allows for the adaptation of pre-trained features to the FMD model (Xiao *et al.* 2019). The freezing process is conducted for the purpose of selectively freezing specific layers within the pre-trained model during training. While some layers remain frozen to retain learned features, others are unfrozen to facilitate updates to the learned features during training (Xiao *et al.* 2019; Wang *et al.* 2023). A lower learning rate of 0.0001 is utilised to fine tune the model to facilitate precision of adjustments made to the FMD model parameters during model compilation (Yi, Ahn and Ji 2020). Furthermore, various data augmentation techniques are applied to assist in mitigating overfitting that occurs. Concluding with fine-tuning the model, callbacks including early stopping, applying a patience of 10 and learning rate scheduling are applied to the training process for the FMD model (Muhammad *et al.* 2022). This can help to stabilise and accelerate the training process (Yi, Ahn and Ji 2020; Salehi *et al.* 2023; Wang *et al.* 2023).

4.2.6.2. Masked Facial Recognition

Fine-tuning the CNN model is conducted through refinements and adjustments made based on specific components of the model. This includes incorporating batch normalisation layers and reducing dropout rates (Ding *et al.* 2019). Through the application of these techniques, overfitting is reduced and model generalisation is improved. In addition, hyperparameters including the learning rate of the optimiser set at 0.0005 are fine-tuned to ensure efficient model training in MFR (Yi, Ahn and Ji 2020). Additional techniques including early stopping with a patience of 10, model checkpoint, and learning rate reduction on plateau are employed for effective training and

optimisation (Muhammad *et al.* 2022). The reduction on plateau dynamically adjusts the learning rate based on the validation loss trend during training to further optimise performance and prevent overfitting of the MFR model (Zhang and Deng 2020).

4.2.7. Prediction and Deployment

In concluding the CNN seven-step model development, the model transitions from conceptualisation, training, validating, testing and concludes with the ultimate practical application for the utilisation of a real-world scenario. The implementation of practical applications are conducted by making predictions on new unseen data as described in the prediction and deployment step.

4.2.7.1. Face Mask Detection

Within the prediction and deployment step for FMD, the trained model is utilised to make predictions based on new and unseen data. This process is conducted by testing the FMD model's performance through assessing its accuracy on a sperate unseen test dataset. This test dataset consists of randomly selected test images based on masked and unmasked face participants. These images are acquired from sources such as different datasets within the two selected FMD datasets. Based on the satisfaction of the model performance (i.e., the model presents accuracy rates of 95% and higher) within the test images, the model is tested in a real-time scenario. In this real-time detection, the process begins with the initialisation of a web camera through a local screen window. This image then goes through a series of steps to detect the input facial image presented to the model in the mask detection frame. If the participant is a wearing mask, a green bounding box pops up around the input face and a label titled *"Mask"* along with the accuracy percentage of the detected mask is displayed on the screen. If the participant is not wearing a mask, a green bounding box appears on the screen titled *"No Mask"* along with the accuracy percentage for the *"No Mask"*. Hence, through leveraging the FMD model, it becomes possible to ascertain whether individuals adhere to the practice of wearing their face masks or not. Therefore, the capability of the FMD model can provide predictive capabilities which can open possibilities for real-time monitoring in public spaces where the use of a face mask is mandatory upon deployment.

4.2.7.2. Masked Facial Recognition

In the prediction and deployment step for MFR, the trained model is utilised to make predictions based on new and unseen test data. This process is conducted by testing the MFR model

performance through assessing its accuracy on a sperate unseen test dataset using randomly selected test images. These images are acquired from sources such as different datasets within the three selected MFR datasets and the *'face mask detection'* dataset (Gurav 2020). The images consist of known and unknown masked participants. In addition, the model is tested in a real-time scenario. In the real-time recognition, the process begins with the initialisation of a web camera through a local screen window. This image then goes through a series of steps to detect the and recognise the input facial image presented to the model in the masked face recognition frame. If the correct identity of the individual is recognised, a green bounding box pops up around the participant's face and a label with the participant's name and the accuracy percentage of the recognised participant is displayed on the screen. If the participant is not recognised, a red bounding box appears on the screen, titled *"Unknown"* and a value of "*0.00*" is set for unknown individuals. Once the model's accuracy and reliability are verified, it can be deployed for practical use, such as in security systems or identity verification applications.

4.3. Materials

In this section the essential materials utilised in this study are described in detail. These materials include a detailed description of the datasets and software libraries used to develop, train, validate and test the proposed FMD and MFR models.

4.3.1. Dataset

In both datasets, there is a diversification of face mask types, participant, genders, skin tones, angles, and the quality of the images range from extreme High Definition (HD) to significantly blur, making the datasets suitable for use.

Within the FMD phase of the research, two datasets are employed. These datasets include the *'face mask detection'* (Gurav 2020) with 5980 images and *'Face Mask Detection 12K images dataset'* (Jangra 2020) with 5000 images, both from the Kaggle repository. A custom dataset is formed for both datasets given the excessive volume of images, a more streamlined selection is deemed appropriate for the study, as the vast quantity exceeded the study's requirements.

In the MFR phase of the research, three datasets are employed. The *'Masked Face Recognition'* (Singh 2023a) dataset contains 465 images, taken from the Kaggle repository. Using the augmentation equation described in *equation 4.1,* $((24 \times 21) + 24) \times 19$, a total of 10032

images are created. Additionally, a dataset titled *'Mask Detection and Masked Face Recognition'* developed by author Ullah *et al.* (2022), contains a total of 500 images. Through the application of the augmentation equation defined in *equation 4.1,* $((10 \times 21) + 10) \times 50$*,* a total of 11000 images are generated. For the *'Mask Detection and Maked Face Recognition'* dataset by author Ullah *et al.* (2022) , a tailored dataset is developed due to the excessive volume of images. This developed selection of images ensure a more focused and relevant sample size for the proposed study, aligning with the specific needs and objectives of the research. Considering the scarcity of publicly accessible datasets, both *'Masked Face Recognition'* and *'Mask Detection and Masked Face Recognition'* datasets are selected due to their availability to the public and the suitability of their sample size required for the proposed study.

Further, a custom dataset is developed to test the performance of the model in a real-time context. This dataset consists of 10 participants that make up 250 images. Using the augmentation equation described in *equation 4.1,* $((25 \times 21) + 25) \times 10$*,* a total of 5500 images are created. The purpose of creating the custom dataset titled *'Custom Real-Time Masked Face Recognition'* is to validate the model's effectiveness in real-life scenarios, ensuring it can recognise individuals accurately and efficiently in real time.

4.3.2. Software Libraries

Software libraries such as TensorFlow, Keras, and OpenCV play a pivotal role in advancing computer vision and deep learning applications. In image processing and video analysis, OpenCV is a commonly used open-source library that enables tasks such as face detection and feature extraction (Bussa et al. 2020). Deep learning models can be developed and trained easily using Keras, a framework built on top of TensorFlow (Agrawal et al. 2021). TensorFlow, a powerful framework, empowers the creation of complex neural networks for various tasks, including face recognition (Jose 2019). TensorFlow, Keras and OpenCV are the main software libraries used to develop the FMD and MFR models.

I. <u>TensorFlow</u>

TensorFlow serves as an open-source machine learning framework. Positioned as the latest second-generation creation from Google (Jose 2019), it finds application in the construction of Artificial Intelligence (AI) models, particularly in the domain of image recognition training. By employing dataflow graphs rooted in numerical computations,

TensorFlow facilitates the translation of intricate data structures into artificial neural networks (Pang, Nijkamp and Wu 2020). Its open-source nature allows for versatile deployment across various devices. Notably, the CNN harnesses the power of TensorFlow for tasks such as facial recognition (Yuan et al.2017). Leveraging this open-source framework aids in the development of the FMD and MFR models (Pang, Nijkamp and Wu 2020). The utilisation of TensorFlow presents several advantages. These benefits encompass the capability to yield greater accuracy results when compared to alternative facial recognition algorithms. Additionally, TensorFlow demonstrates commendable computational performance, excels in generating intricate graph visualisations, and provides compatibility with a variety of devices due to its open-source nature. In complex settings, TensorFlow therefore exhibits a heightened level of robustness, suitable for aiding in the development of the FMD and MFR models (Yuan *et al.* 2017; Pang, Nijkamp and Wu 2020).

II. Keras

Keras serves as a pivotal tool in the realm of CNNs. Functioning within the Python programming language, Keras emerges as a prominent open-source library, for the construction of neural networks (Agrawal et al. in 2021). The framework seamlessly integrates with the TensorFlow framework, supporting its open-source nature to allow the library to support the use of back-ends, since it operates with high level computations (Agrawal et al. in 2021). Hence, using Keras with TensorFlow provides efficient optimisation using callbacks such as early stopping and reduceLR on plateau (Dürr, Sick and Murina 2020). The utilisation of this framework provides numerous advanatges in its adoption. Classification tasks in image detection and recongition are simplified with it since the library provides numerous pre-trained models (Chen *et al.* 2018). Feature extraction, prediction and fine-tuning are conducted using the pre-trained models as the framework contains easy-to-learn code features which are bound with a user-friendly API that is simple to implement in deep and machine learning algorithms (Nguyen *et al.* 2019). Keras allows for the processing of large amounts of image data, since it utilises multiple GPUs to train the model. Hence, it supports data parallelism throughout the model (Sergeev and Del Balso 2018; Dürr, Sick and Murina 2020). In this research study, Keras plays a vital role in implementing various deep learning functionalities crucial for the development

of the FMD and MFR. Through leveraging the capabilities of Keras, pivotal CNN functions are constructed and fine-tuned. These functions include one-dimensional convolution layers, max pooling, the flattening process, dense layers, batch normalisation, dropout, model checkpoint and aids in the implementation of augmentation techniques.

III. OpenCV

OpenCV stands as a prominent open-source platform tailored for the domain of computer vision applications for machine and deep learning models. The library extends itself to tasks such as the identification and retrieval of images from expansive databases, object recognition, and the facilitation of three-dimensional (3D) object modelling (Bussa et al. in 2020). With over 2500 algorithms, it offers a comprehensive suite of computational tools tailored to diverse visual processing tasks, provides extensive data storage capabilities, and automated identification (Dhawle, Ukey, and Choudante, 2020). Within the context of this research study, the OpenCV library plays a crucial role in various aspects of model development. The OpenCV library facilitates the preprocessing phase of the input dataset images by enabling operations such as reading, resizing, and converting colour spaces (Bussa et al. in 2020; Dhawle, Ukey, and Choudante, 2020). These preprocessing operations are essential for preparing data for training the FMD and MFR models. In addition, OpenCV is utilised for real-time masked face detection, recognition, and localisation. It assists in providing the necessary inputs required for the subsequent processing steps of the model (Jagtap et al. 2019). The library aids in visualising the results of the detected and recognised faces by drawing colour bounding boxes around detected faces whilst overlaying crucial information such as the confidence scores and classification labels within the video feed, real-time detection (Jagtap et al. 2019; Bussa et al. in 2020). Furthermore, OpenCV enables the augmentation of dataset images, enhancing the diversity and robustness of the training data, contributing to the overall performance of the FMD and MFR models.

4.4. Ethics Consideration

The research proposal of this study has been approved with ethics category one. Due to the use of datasets and minimal risk to humans that the research proposal was approved with no need for further ethical clearance approval (*Annexure B*).

4.5. Chapter Summary

Within this chapter, the required methods and materials utilised for the model development were presented to offer a comprehensive understanding based on the development of the FMD and the MFR models. A detailed process flow was provided to aptly visualise the flow of the aforementioned models, thus gaining insight into the implementation of both the models. In addition, all seven steps of the CNN model development are detailed with the respective descriptive information on the structural methodology of each model. Furthermore, within the seven-step model, a detailed description is provided based on the performance metrics (i.e., the confusion matrix and evaluation metrics) which are used to evaluate the FMD and MFR models. In concluding this chapter, the research materials and software libraries, encompassing datasets and frameworks employed in this study, are thoroughly examined, and discussed. Due to this thorough methodology, the model is not only resilient but also suitable for practical applications. In the subsequent chapter, detailed experiments and results are presented and analysed to provide comprehensive insights based on the efficacy of the developed models.

CHAPTER FIVE: RESULTS AND ANALYSIS

Chapter five delves deeply into the results and analysis of the experiments conducted based on the developed Convolutional Neural Network (CNN) for the Face Mask Detection (FMD) and Masked Facial recognition (MFR) models. This chapter, therefore, provides an in-depth exploration into the critical phase of model evaluation, assessing its effectiveness in detecting the presence and non-presence of face masks and recognising the identity of masked faces. By harnessing two distinct datasets for FMD and three datasets for MFR, this study navigates through a meticulous exploration of experimental results. The conducted experiments highlight scenarios that include varied lighting conditions, diverse facial angles, and the incorrect use of a face mask. In addition, the research delves into the scrutiny of confidence levels, examining their dynamics across all models in the training and testing phases, as well as in simulated and real-time scenarios. This is conducted for the purpose of detecting and recognising participants in the FMD and MFR segment of the research. Hence, a qualitative and quantitative analysis of results are presented in this chapter.

5.1. Model Development and Experiments

The models were developed on an HP ProBook 450 G4 laptop (x64-based PC) with 16 gigabytes of random-access memory and 232 gigabytes of storage space, utilising both the Intel R High Definition (HD) graphics 620 integrated graphics processing unit and central processing unit. The experimental development process involved executing the constructed models within the Anaconda (Aryotejo *et al.* 2023) Integrated Development Environment (IDE). The models were developed and built using the Jupyter notebook open-source web application, using Microsoft Edge as the web platform (Granger and Pérez 2021). The initial phase of the experimental process involved training the models with various images. In the FMD phase, the images contain participants that wear a face mask and participants that do not, in order to build a robust model that is flexible and able to easily identify individuals that wear masks. In the second phase, within context of the MFR model, a multitude of masked and non-masked images containing multiple participants for each dataset are trained to accurately recognise a participant whilst their face mask is on.

5.2. Face Mask Detection Model

The FMD model is designed and developed to automatically identify whether individuals are wearing face masks or not in real-time. Moreover, the model can discern instances where a person wears their face mask incorrectly or employs their hands to cover their mouth, categorising such scenarios as a *"No Mask"* detection. This can help to enforce safety protocols and public health measures during situations that require the use of a face mask. *Section 5.2.1.* to *Section 5.2.4.,* provide an illustrative overview based on the development and results implemented for the required FMD model. The FMD model is experimented on with the *'face mask detection'* (Gurav 2020) and *'Face Mask Detection 12K images'* (Jangra 2020) datasets. The analysis and results begin with the dataset visualisation and summary. The model results derived from the training phase is then depicted and described in detail. In the last stage the model performance is discussed, highlighting the evaluation metrics applied to determine the effectiveness of the proposed model when compared to the existing research literature findings. The section concludes with a summary based on the FMD model documented outcomes.

5.2.1. Face Mask Detection Datasets

In the context of *'face mask detection'* and *'Face Mask Detection 12K images'* datasets, this study explores FMD, presenting a comprehensive experimentation and the ensuing results based on the respective datasets. As displayed in *Figure 5. 1* and *Figure 5. 2,* the datasets contain multiple variations of images. Both datasets consists of extremely blurry, HD, various light intensities, diverse facial angles, several images with cropped regions, and a diverse range of ethnicities. All these images are utilised to aid the development of a generalised and robust model.

5.2.1.1. The 'face mask detection' Dataset

This dataset is composed of a total of 5980 images. The images are compartmentalised based on two folders labelled *"Mask"* and *"No Mask."* Both the *"Mask"* and *"No Mask"* classes contain a balanced and equally distributed number of images containing 2990 in each folder.

Figure 5. 1: Sample Images of the 'face mask detection' Dataset (Gurav 2020)

5.2.1.2. The 'Face Mask Detection 12K images' Dataset

In this dataset, a total of 5000 images are included. The *"Mask"* and *"No Mask"* classes both contain a balanced and equally distributed number of images with 2500 images in each folder.

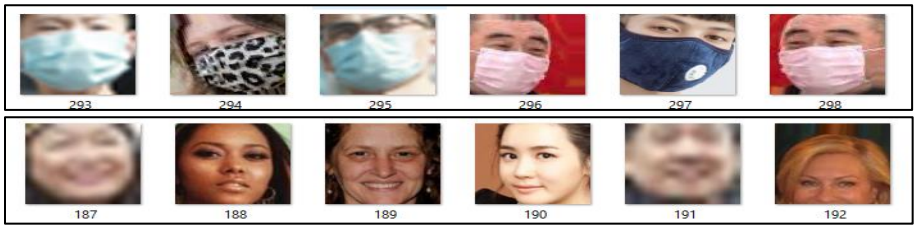


Figure 5. 2: Samples Images of the 'Face Mask Detection 12K Images' Dataset (Jangra 2020)

Investigating the outcomes of training the FMD model provides critical insight into their performance and efficacy. The next section explores the results obtained from training the FMD model using the aforementioned datasets in identifying the presence and non-presence of face masks on individuals.

5.2.2. Face Mask Detection Model Training Results

The dataset images are trained using a variation of CNN known as the MobileNetV2 architecture, optimising parameters to enhance accuracy in predicting or classifying new, unseen data to perform classification proficiently.

5.2.2.1. The 'face mask detection' Dataset

The distribution of images which follow an 80/10/10 split, with 4784 images allocated for training the model, 598 images utilised for validation during hyperparameter tuning, and 598 images reserved for assessing the model's accuracy in classification testing on unseen test data. As depicted in *Annexure C, Figure 1,* the representation demonstrates the model's accuracy throughout both the training, validation and testing process.

The model accuracy is defined as the mean accuracy across individual batches of training data. The metric "val_accuracy" is precisely reflective of the accuracy based on the predictions within the validation data batch. The "val_loss" exhibited in the model diagram for each epoch quantifies the instances where predictions do not align with the values in the validation data batches. This loss metric reveals discrepancies identified within the model across various batches of data. The last epoch displayed an apt validation accuracy of 1.0000 and a validation loss of 0.0024.

Once the model has been trained using this dataset, it is effectively evaluated based on the performance of the deep learning model within the test portion of the dataset. This evaluation is performed to calculate the average loss and accuracy over the entire unseen test dataset. The model produced an overall exceptional accuracy rate of 100%. The model in conjunction with the exceptional accuracy rate also produced an overall low loss of 0.0012.

Upon completion of the trained, validated and tested images, through the utilisation of plotted graphs, a demonstration based on the evolution of key metrics across the epochs during the training process is visualised.

During training, machine or deep learning models are evaluated using training accuracy and training loss metrics. This indicates the model's fit to the training data. Training accuracy represents the percentage of correctly predicted labels, while training loss quantifies the disparity between the model's predictions and the actual targets. Decreasing training loss signifies improved accuracy and serves as an optimisation objective. Visual representations, such as the graph depicted in *Figure 5. 3*, offer insights into the accuracy and loss trends that occur during the training process. Furthermore, validation accuracy and loss are assessed on a distinct subset of the dataset to facilitate hyperparameter tuning, ensuring that the model avoids overfitting and achieves robust generalisation. Conversely, test accuracy and the test loss, evaluated on a separate test portion of the dataset, inherently assess the model's performance on the unseen portion of the test data. The test accuracy therefore measures the percentage of accurately predicted labels, while test loss quantifies the deviation between predictions and targets. These metrics, crucial for gauging real-world performance, aid in identifying potential overfitting issues and optimising model generalisation (Akhtar and Ragavendran 2020).

Based on the training, validation and test results obtained during the epoch iteration, there is a notable improvement in model performance over the epoch iteration process. The training accuracy begins with a value of 56% and steadily increases to a value of 99.44%, indicating effective learning. Similarly, the validation accuracy begins with 55.02% and steadily increases to a value of 100%. The test data, performed on the unseen images, provided a test accuracy of 100%. This highlights the model's ability to generalise on unseen data effectively. In the loss function, the training loss decreases from 0.8164 to 0.0214 for training and 0.5753 to 0.0024 for validation. Within the unseen test data, a loss of 0.0012 was observed.

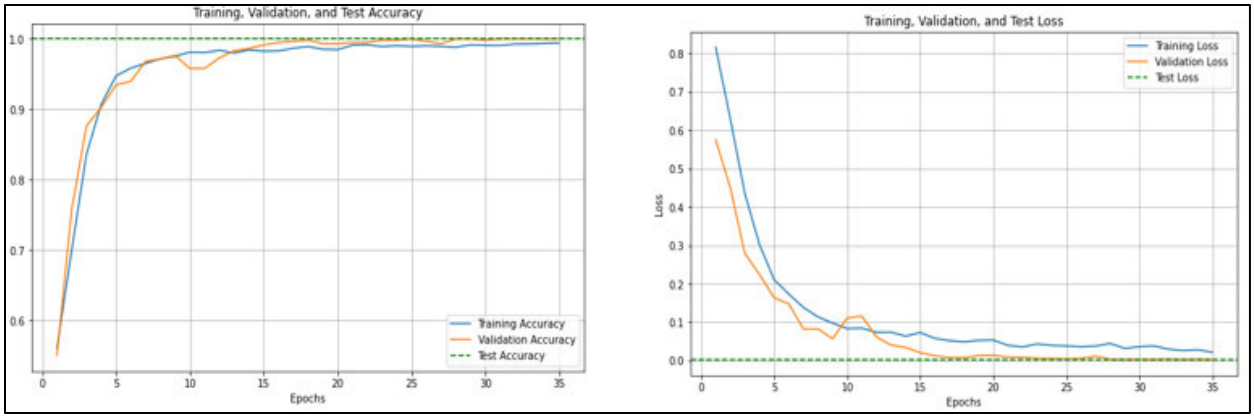


Figure 5. 3: Plotting the Epochs of the 'face mask detection' Dataset

This indicates efficient convergence and therefore a strong predictive capability of the model is displayed. In addition, the model exhibits no signs of overfitting, as indicated by consistently improving validation accuracy and with a minimal disparity between training and validation loss in the graph visualisations.

5.2.2.2. The 'Face Mask Detection 12K images' Dataset

The distribution of images following the aforementioned split consists of 4000 images allocated for training, 500 images for validation and 500 images reserved for assessing the model's accuracy in classification testing on the unseen data. The accuracy of the model that occurred during the epoch iteration training, validation and testing process is depicted in *Annexure C, Figure 2*. The last epoch displayed a validation accuracy of 1.0000 and a validation loss of 0.0020.

Upon completion of the trained model using this dataset, the deep learning model is evaluated based on the model's performance on the unseen test dataset. The model produced an overall accuracy of 100%, emphasising the model's remarkable performance. Furthermore, the model produced an overall loss of 0.0013, demonstrating insignificant discrepancies and highlighting its exceptional performance.

The training, validation and testing accuracy, and training, validation and testing loss are illustrated in *Figure 5. 4*. The provided training, validation and testing results reveal a steady progress in both loss and accuracy metrics during the respective epoch iteration process. The FMD model demonstrates a moderate performance and progressively increases through each epoch iteration, with an initial training accuracy of 53.60% to 99.45% and a validation accuracy of 69.00% to

100%. The test data, performed on the unseen images, provided a test accuracy of 100%. In the loss function, the training loss decreases from 0.8853 to 0.0260 for training and 0.5504 to 0.0020 for validation. The test loss produced an outcome of 0.0013. Hence, there is a consistent increase in accuracy and decrease in the loss on both the training and validation sets throughout the epochs. It therefore suggests that no overfitting has occurred.

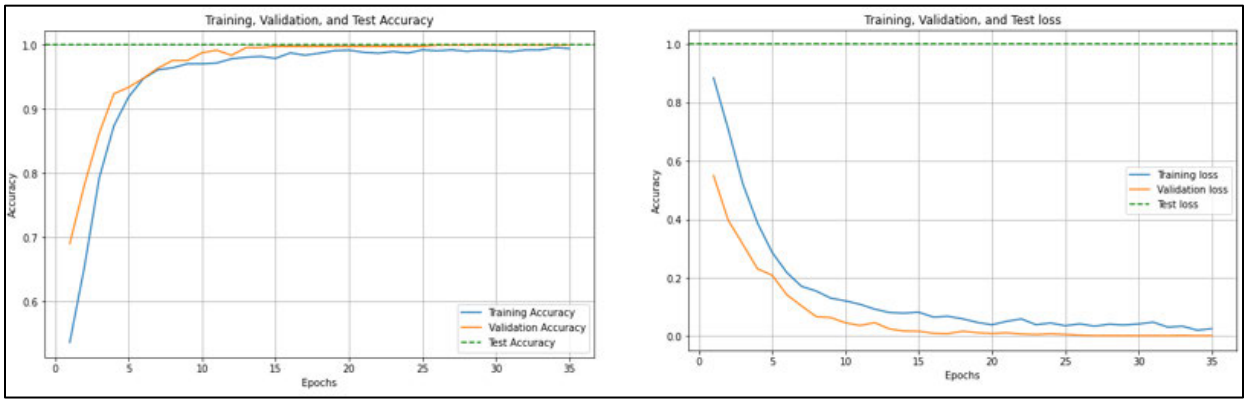


Figure 5. 4: Plotting the Epochs of the 'Face Mask Detection 12K images' Dataset

The evaluation of model performance through the confusion matrix and evaluation metrics offers valuable insights into its effectiveness and robustness of the model. The results of these metrics based on model performance are illustrated both quantitatively and qualitatively within the *model performance* section.

5.2.3. Face Mask Detection Model Performance

The performance of the model is evaluated on the *'face mask detection'* and the *'Face Mask Detection 12K images'* datasets using the confusion matrix, accuracy, precision, recall, and the f1-score.

5.2.3.1. Model Performance - 'face mask detection' Dataset

The confusion matrix depicted in *Table 5. 1,* provides an illustrative overview of the model. In this matrix, there are 299 *True Positive (TP)* images, indicating that 299 out of a total of 299 test images per class are correctly and accurately representative of the *"Mask"* class. In the *False Positive (FP)* class, there are zero (0) instances where the model falsely classified negative *"No Mask"* instances as positive *"Mask"* instances. In the third block known as the *False Negative (FN)* class, there are zero (0) instances where the model falsely classified positive *"Mask"* instances as negative *"No Mask"* instances. This overview depicts 299 *True Negative (TN)* images, indicating

that 299 out of a total of 299 test images per class are correctly and accurately representative of the *"No Mask"* class.

Table 5. 1: Confusion Matrix on the 'face mask detection' Dataset

		Actual	
		Mask	**No Mask**
Predicted	**Mask**	TP (299)	FP (0)
	No Mask	FN (0)	TN (299)

By adding both the *TP* (299) and *TN* (299) classes and dividing it by the total number of testing images (598), it provides insight into whether the model is overfitting or not. Performing this calculation results in a total of 100%. Hence, there is no overfitting that occurred since the unseen test data provides a score of 100% and the confusion matrix calculation is also 100%.

In assessing model performance, a comparison of the proposed model against the existing literature is performed using evaluation metrics such as the accuracy, precision, recall, and f1-score. A quantitative comparison is therefore performed based on the existing literature models that use the same *'face mask detection'* dataset to train and assess model performance as depicted in *Table 5. 2*. Referring to *Annexure A*, the entire study result for each model is found based on the corresponding author name.

Table 5. 2: Performance Evaluation on the 'face mask detection' Dataset

Study	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Proposed Study	100	100	100	100
(Bania 2023)	99.97	99.97	99.97	99.97
(Habib *et al.* 2022)	99.98	99.96	99.97	99.97
(Benifa *et al.* (2023)	99.00	98.00	99.00	99.00
(Guo *et al.* 2022)	96.70	96.30	92.50	94.36
(Goyal et al. 2022)	98.00	98.00	98.00	98.00

Based on the comparison performed in *Table 5. 2,* it is evident that the proposed model outperformed the existing models, attaining a 100% score across the aforementioned evaluation

metrics. In addition, it is important to note the identified limitations within each of the existing studies that have been effectively addressed within this research. Within these studies the following limitations are noted.

- In the study by author Bania (2023) , the model lacks the ability to identify instances when individuals wear face masks incorrectly. The outcomes solely reflect individuals presenting frontal face images in real-time. Hence, the model does not cater for images at varied angles.
- Within study by Habib *et al.* (2022), the model focuses on detecting frontal face images, lacking the capability to discern different face mask positions, including instances with no mask or incorrect mask placement.
- Study three by Benifa *et al.* (2023) highlights the model's responsiveness is highly influenced by the spatial orientation of the camera. In addition, the model's potential is constrained as it relies on larger datasets for training, and the acquired weights are not fully optimised for real-world applications without further enhancement. The model required a considerable number of 100 epochs to train in order to produce an accuracy of 99%.
- In the study by Guo *et al.* (2022), the model only achieved an accuracy of 96.70%, indicating the lowest accuracy in comparison to the rest of the models. The current dataset, while extensive, poses a limitation as it lacks heavily obscured and half-face images. Therefore, the model produces a considerable number of false positives. In addition, this model required a total of 100 epochs to train the model efficiently.
- The last study by author Goyal et al. (2022), utilises a selection of images from the *'face mask detection'* dataset to create a custom dataset for model training and testing. In this study, the model only achieved score of 98% throughout the aforementioned metrics in comparison to the proposed model that achieved 100%. In addition, the model is incapable of detecting improper mask usage, thus creating an increased number of false positives in the improper mask usage scenario.

Based on *Figure 5. 5*, a graph-oriented visualisation of the proposed model is depicted for the *'face mask detection'* dataset. Thus, it provides a score of 100% across the aforementioned evaluation

metrics when compared to the existing models that utilised the '*face mask detection*' dataset identified in the existing literature.

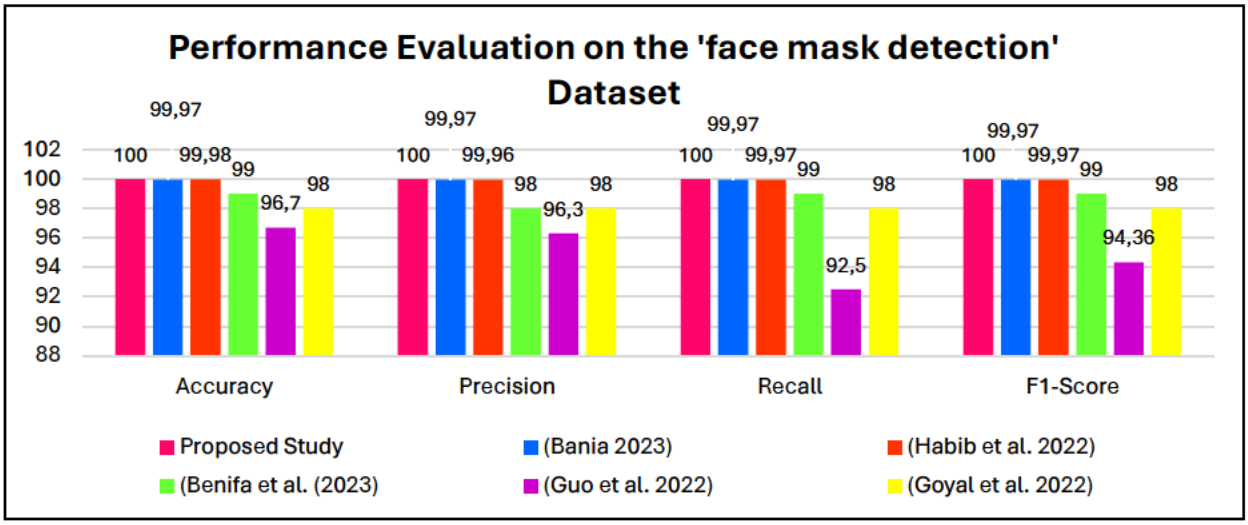


Figure 5. 5: Performance Evaluation on the 'face mask detection' Dataset Graph

Using sample images from the *'Face Mask Detection 12K images'* dataset, the model performance is further assessed. The predicted results based on a set of test images are illustrated and described aptly in *Table 5. 3* and *Figure 5. 6*.

Table 5. 3: Results of the 'face mask detection' Dataset Sample Prediction

Label	Description	Accuracy Percentage (%)
A	Frontal face and no mask with glasses on.	100
B	No mask, blurry image and a slight facial angle.	100
C	Mask on, highly blurry image and head at an extreme left angle.	100
D	Mask on, extremely blurry image with a slight facial angle and glasses on.	100
E	Augmented image, no mask, head at a slight right angle with reduced brightness and distorted colour orientation.	100
F	Augmented image, no mask, highly blurred image with noticeable distortion and a slight right angle of the head.	100
G	Augmented image, mask on, highly blurred image with noticeable distortion and a slight right angle of the head.	100
H	Augmented image, mask on, slight right angle of the head with reduced brightness and distorted colour orientation.	100

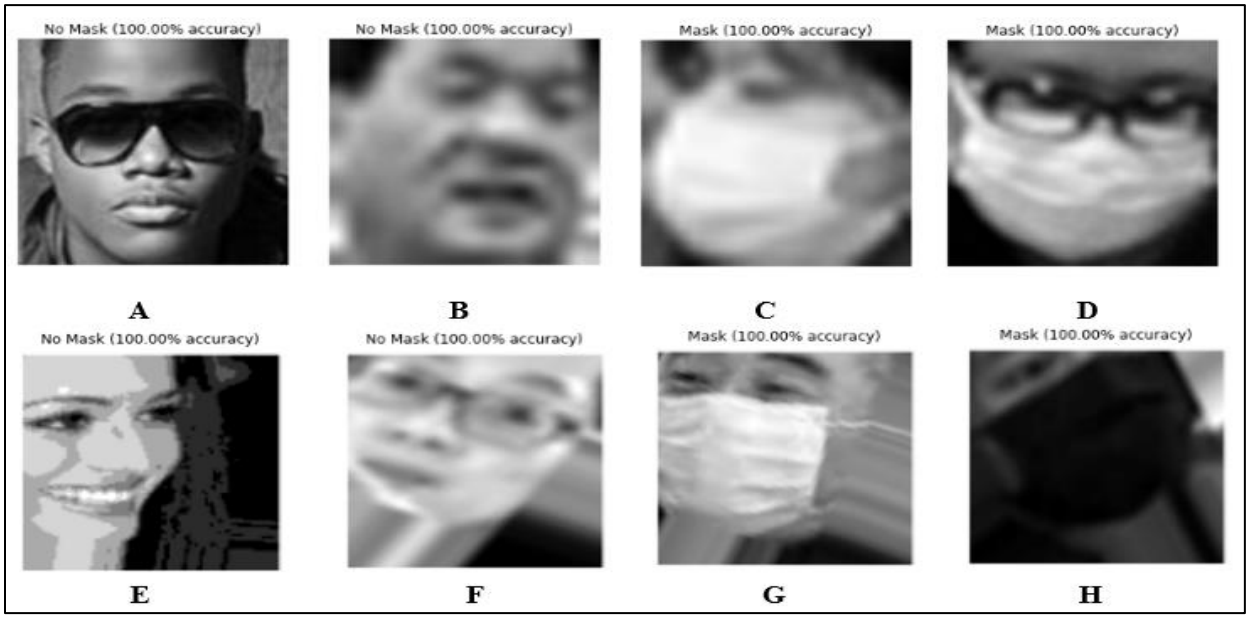


Figure 5. 6: 'face mask detection' Dataset Sample Prediction (Jangra 2020)

As a result, the model consistently achieved a 100% accuracy score across all sample image scenarios presented to the model, demonstrating the robustness of the FMD model.

5.2.3.2. Model Performance - 'Face Mask Detection 12K images' Dataset

The confusion matrix depicted in *Table 5*. 4, provides an illustrative overview of the model. The results reveal that there are 250 *TP* images, indicating 250 out of 250 total test images per class are correctly and accurately representative of the *"Mask"* class. In the *FP* class, there are zero (0) instances where the model falsely classified negative *"No Mask"* instances as positive *"Mask"* instances. In the third block known as the *FN* class, there are zero (0) instances where the model falsely classified positive *"Mask"* instances as negative *"No Mask"* instances. This overview depicts 250 *TN* images, indicating that 250 out of 250 total test images per class are correctly and accurately representative of the *"No Mask"* class.

Table 5. 4: Confusion Matrix on the 'Face Mask Detection 12K images' Dataset

		Actual	
		Mask	**No Mask**
Predicted	**Mask**	TP (250)	FP (0)
	No Mask	FN (0)	TN (250)

By adding both the *TP* (250) and *TN* (250) classes and dividing it by the total number of test images (500), it provides insight into whether the model is overfitting or not. Performing this calculation results in a total of 100%. Hence, there is no overfitting since the unseen test data provided a score of 100% and the confusion matrix calculation is also 100%.

A comparison of the proposed model using the *'Face Mask Detection 12K images'* dataset against the existing literature models that utilise the same dataset are performed based on the aforementioned evaluation metrics depicted in *Table 5. 5.*

Table 5. 5: Performance Evaluation on the 'Face Mask Detection 12K images' Dataset

Study	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Proposed Study	100	100	100	100
(Fazeli Ardkeen, Tale and Parseh 2023)	99.02	99.00	99.00	99.00
(Zhang 2021)	98.70	98.00	98.00	98.00
(Chakma *et al.* 2022)	99.46	99.73	99.19	99.46
(Arora, Gupta and Sridhar 2022)	95.00	98.00	95.00	96.50

Based on the comparison performed in *Table 5. 5,* the proposed model outperformed the exiting literature models, attaining a score of 100% throughout the aforementioned evaluation metrics. In performing this evaluation, the following limitations are noted in the existing literature studies.

- The study by author Fazeli Areana, Tale and Parseh (2023), noted that the model required a total of 100 epochs to run. In addition, the model's computational demands are significant, driven by its utilisation of a dataset comprising 12000 images. Moreover, the inclusion of an additional 12000 images through data augmentation further heightened the computational complexity of the model. The model also lacks the capability to detect face masks in real-time. Furthermore, the model identified a considerable number of false

negatives and positives in confusion matrix, indicating that the model struggles with accurately distinguishing between the respective classes.

- In the study by Zhang (2021), the model achieved an average score of 98% throughout the evaluation metrics and 98.70% in the accuracy of the model, attaining a considerably lower score than the proposed model. In addition, the model displayed a considerable number of false negatives and positives in the confusion matrix, implying that it faces challenges in accurately differentiating between the respective classes. Furthermore, the study highlights the need to enhance storage efficiency and overall speed in the detection process, indicating a substantial demand for computational resources.
- Within the study by author Chakma *et al.* (2022), it is evident that the proposed model achieved a higher set of results in comparison to the existing developed model. In addition, the model's effectiveness is limited to frontal face images, as demonstrated by the real-time detection results presented in the research study.
- Based on the results obtained in the study by author Arora, Gupta and Sridhar (2022), the model displayed poor scores in all the aforementioned evaluation metrics. This indicates that the model requires significant improvement. Furthermore, the model required a significant amount of computational resources indicating the computational complexity identified within the model.

Based on *Figure 5. 7*, a graph-oriented visualisation of the proposed model in comparison to the existing literature models using the *'Face Mask Detection 12K images'* dataset is depicted. Thus, it provides a score of 100% across the aforementioned evaluation metrics when compared to the existing models in the literature that utilised the same dataset.

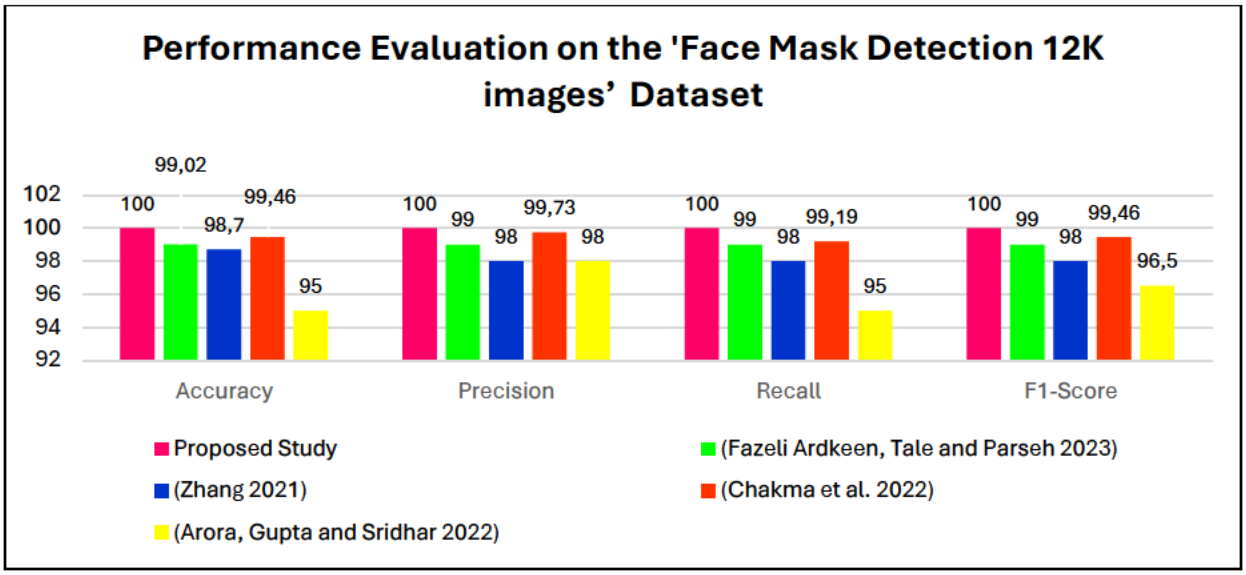


Figure 5. 7: Performance Evaluation on the 'Face Mask Detection 12K images' Dataset Graph

Using sample images from the *'face mask detection'* dataset, the model performance is further assessed. The predicted results based on a set of test images are illustrated and described aptly in *Figure 5. 8* and *Table 5. 6*.

Table 5. 6: Results of the 'Face Mask Detection 12K images' Dataset Sample Prediction

Label	**Description**	**Accuracy Percentage (%)**
A	Frontal face and no mask with a baseball cap on.	100
B	No mask, blurry image and head at an angle with glasses on.	100
C	Mask on, highly blurry image and head at a slight angle with glasses on.	100
D	Mask on, highly blurry image and head at an extreme left angle.	100
E	Augmented image, no mask, head at an extreme right angle with noticeable image distortion.	100
F	Augmented image, no mask, head at an extreme left angle, slightly blurred image with noticeable distortion, reduced brightness and distorted colour orientation.	100
G	Augmented image, mask on, highly blurred image, head at a slight right angle with noticeable distortion and glasses on.	100
H	Augmented image, mask on, frontal face image with reduced brightness and distorted colour orientation.	100

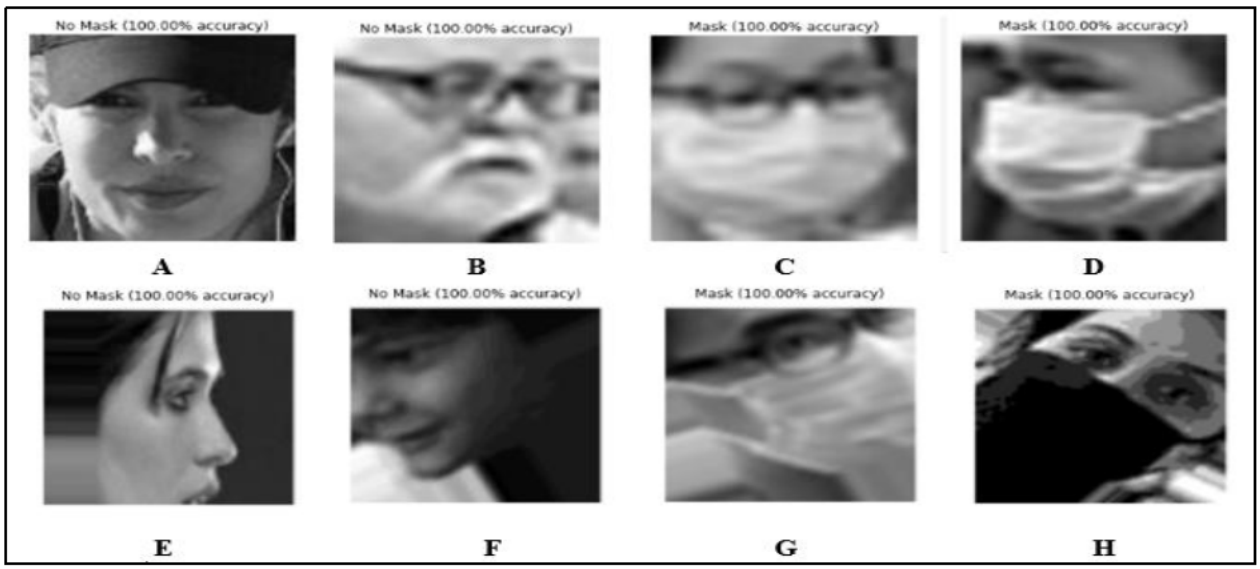


Figure 5. 8: 'Face Mask Detection 12K images' Dataset Sample Prediction (Gurav 2020)

Therefore, the model achieved an accuracy score of 100% in every sample image scenario presented to it, indicating the robustness of the proposed FMD model.

In the next subsection, a quantitative comparison is performed between the proposed model and the top three performing models in the existing literature that utilise the MobileNetV2 architecture to build a FMD model.

5.2.3.3. Model Performance - MobileNetV2 Model

The accuracy, precision, recall and f1-score are aptly discussed, and a comparison visualisation is depicted in *Table 5. 7*. Referring to *Annexure A*, the entire study result for each of the top three performing models can be found based on the corresponding author name. The model comparison is performed based on the mean score of the *'face mask detection'* and *'Face Mask Detection 12K images'* datasets on the accuracy, precision, recall and f1-score combined. The model, therefore displayed a mean score of 100% throughout the aforementioned evaluation metrics.

Table 5. 7: Performance Evaluation on the MobileNetV2 Architecture Model

Study	**Accuracy**	**Precision**	**Recall**	**F1-score**
Proposed MobileNetV2 Study (face mask detection and Face Mask Detection 12K images datasets)	100	100	100	100
(Talahua *et al.* 2021) (Custom dataset)	99.65	100	100	100

(Habib *et al.* 2022) (FMD by Omkar Gurav , FM by Oumina, Makhi, and Hamdi, and the RMFR datasets)	99.98	99.96	99.97	99.97
(Ilyas and Ahmad 2022) (Custom dataset that composed of random public repositories)	100	99.90	99.90	99.90

Based on the comparison of evaluation metrics within *Table 5. 7,* it is evident that the proposed model outperformed the top three performing models that utilised the MobileNetV2 architecture within the existing literature, attaining a mean score of 100%. Furthermore, it is important to note the identified limitations within each of the existing studies that have been effectively addressed within this research.

- In the study by author Talahua *et al.* (2021), the model presented difficulty in the detection of certain faces when individuals wore a face mask. The second limitation noted was the model's inability to detect masked individuals from varied angles presented, as it only worked on frontal face images. The model also experienced decreased accuracy when presented with adverse lighting conditions. In addition, the model required a significant amount of time for processing and training the images as the model demanded a total of 10 hours in training time. Hence, computational complexity was noted within this model.
- Within the study by Habib *et al.* (2022), the model focused on detecting frontal face images, lacking the capability to discern different face mask positions, including instances with no mask or incorrect mask placement.
- In the study by authors Ilyas and Ahmad (2022), the model displayed a significant score across the aforementioned metrics, however the proposed model still outperformed the existing model with 100% scored across the evaluation metrics. In the existing study, the focus is solely on the research and development of a FMD model and does not address a solution for MFR.

Therefore, the proposed FMD model attained exceptional evaluation metric scores when compared to the existing literature identified. This highlights the proposed model's ability to discover patterns and conduct accurate classifications on both test sample images and real-world

applications as illustrated and described in the *model performance* section and the *real-time detection for face mask detection* subsection.

Based on *Figure 5. 9*, a graphical visualisation of the proposed model is depicted below. The graph consists of a combination of both the *'face mask detection'* and *'Face Mask Detection 12K images'* datasets. Thus, it provides a mean score of 100% across the aforementioned evaluation metrics when compared to the top three performing MobileNetV2 models identified in the existing literature.

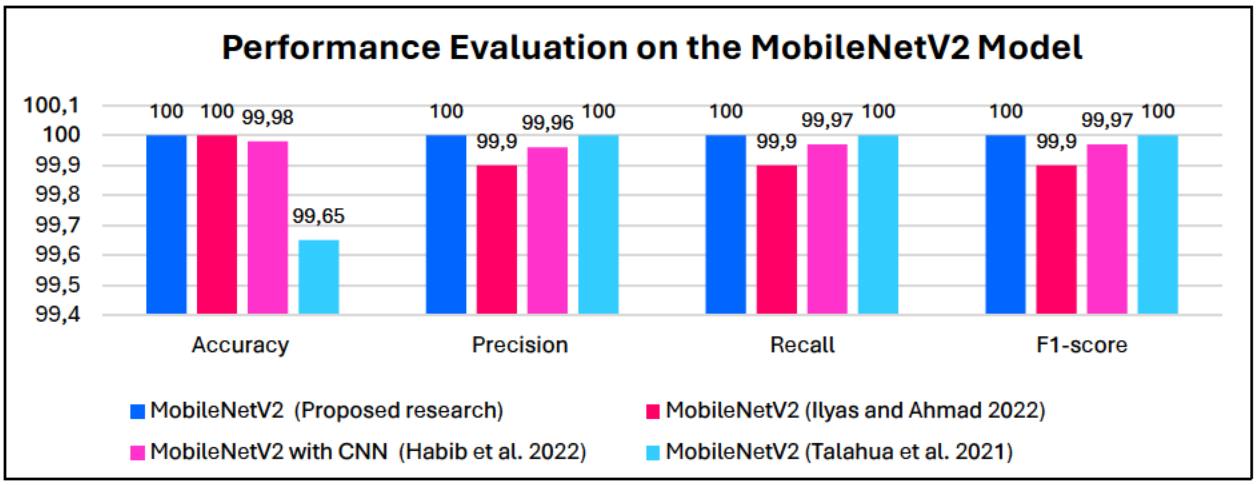


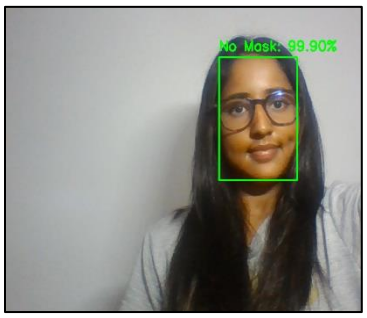
Figure 5. 9: Performance Evaluation on the MobileNetV2 Architecture Model Graph

Presented below is a series of real-time FMD demonstrations. These scenarios demonstrate the model's performance in identifying the presence, non-presence, and improper use of face masks in various real-time scenarios, highlighting its practical application and reliability.

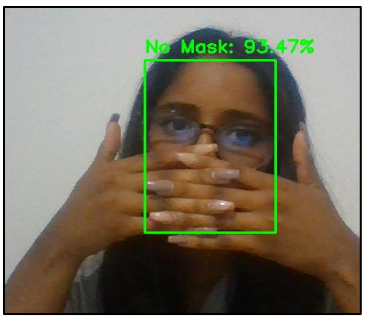
5.2.3.4. Real-Time Detection for Face Mask Detection

The series of images below present an illustration based on real-time FMD. In labels *A to F,* the detection is conducted using a normal standard home lighting condition of 50 lux. In *G to L* , the lights are completely switched off and the same experiments are conducted. Based on labels *E and F,* extreme variation in angles are presented on both the left- and right-hand side with the normal standard lighting conditions applied. In labels *K and L,* extreme variation in angles are presented on both the left and right-hand side with the lights completely switched off. Within label *B*, the scene is illuminated, while in *H*, the lights are entirely extinguished, and in *I,* there is fraction of illumination while the lights are still completely extinguished through the utilisation of a dimmed (i.e., lowest setting) iPhone 11 torch light. All three scenarios display the participant's hands

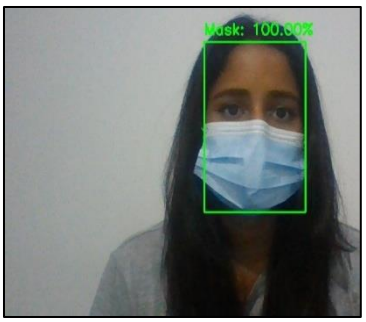
covering the facial region in place of a face mask. In this scenario the model is still able too accurately identify that it is not a face mask and labelled the outcomes as *"No Mask."* Lastly in label *D,* the model is able to accurately identify that even though the participant had a mask on, it is not placed properly on the required facial region rendering the output as *"No Mask."*



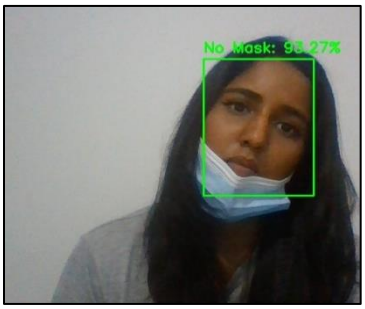
A



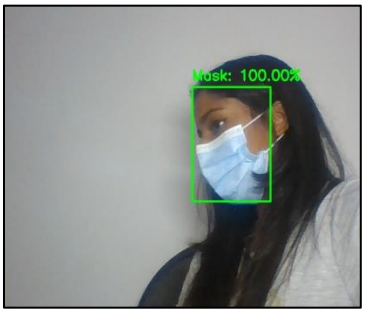
B



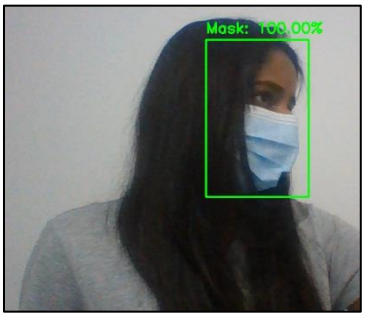
C



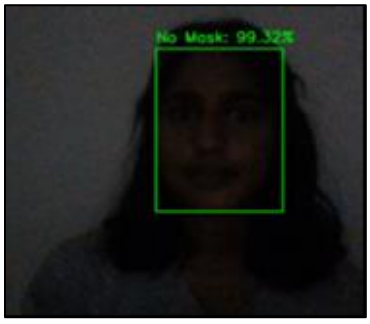
D



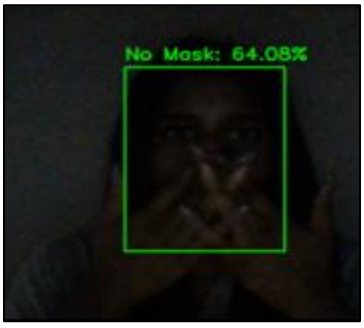
E



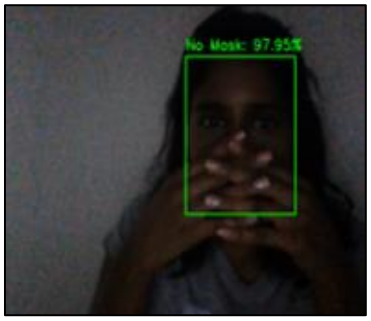
F



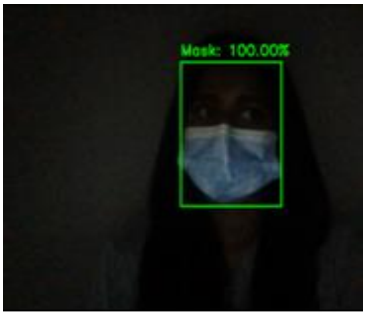
G



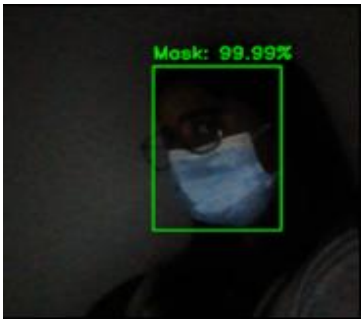
H



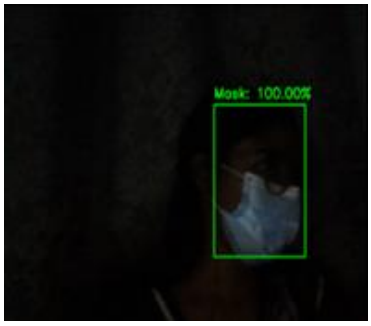
I



J



K



L

Table 5. 8 provides a description of each image based on the series of images labelled *A to L* to provide an in-depth description with reference to the experiments conducted for FMD below.

Table 5. 8: Results of Real-Time Detection for FMD

Label	Description	Accuracy Percentage (%)
A	No mask.	99.90
B	No mask, hands covering face as a mask.	93.47
C	Mask on, frontal face.	100
D	Incorrect use of a face mask.	93.27
E	Mask detection, left angle.	100
F	Mask detection, right angle.	100
G	No mask in the dark.	99.32
H	No mask, hands covering face as a mask in the dark.	64.08
I	No mask in the dark, frontal face with slight illumination.	97.95
J	Mask on in the dark, frontal face.	100
K	Mask detection in the dark, left angle.	99.99
L	Mask detection in the dark, right angle.	100

During both testing on sample images and in real-time detection, the model displayed significant results in its ability to detect the presence and non-presence of a face mask on a participant. In the real-time detection phase, the model produced accuracy results that ranged from a perfect 100% score to the minimum score of 64.08% obtained in extreme dimmed lighting conditions. The model however is still able to accurately identify the non-presence of a mask since the participant employed their hands to cover the facial region. This further reinforces that the model is able to accurately detect the incorrect usage of a face mask in varied lighting conditions despite the necessary facial regions being covered in place of a face mask. In addition, the model is able to detect the presence or non-presence of a face mask whilst the participant had objects such as a baseball cap or the participant's hair down slightly covering the facial structure in certain instances as well as glasses covering the eyes and a portion of the nose region.

5.2.4. Face Mask Detection Summary

Within in the current developed FMD model, the overall mean accuracy score achieved on the unseen test data utilising the two aforementioned FMD datasets is calculated as $100 + 100 = \frac{200}{2} = 100\%$. In addition, the combined average loss achieved on the unseen test data for the FMD model is calculated as $0.12 + 0.13 = \frac{0.25}{2} = 0.125$. The model achieved a total mean accuracy of 100% and a total mean loss of 0.125. The proposed model therefore outperforms the exiting models outlined in the FMD portion of the existing literature, highlighted in the meta-analysis, *Annexure A*. In addition, the developed model displayed an outstanding performance and achieved exceptional accuracy rates throughout real-time scenarios and application testing of the model. The model when run on each of the aforementioned datasets, took a total of one hour each to run with low computational complexity given the laptop hardware described above. Furthermore, the model possessed the ability to detect the presence, non-presence, or incorrect usage of a face mask despite being presented with dimmed lighting conditions, diverse facial angles, distorted blurry images, object obstructions (i.e., eye glasses, baseball cap and hair covering slight regions of the face) and diverse ethnicities as depicted in the *model performance* section.

5.3. Masked Facial Recognition Model

The MFR phase is designed and developed to enable efficient and secure access management, control and monitoring in high-security environments, during pandemics or instances that require the use of a face mask. *Section 5.3.1 to Section 5.3.4* provide an in-depth and illustrative overview of the analysis and results for the required MFR phase. The MFR model is experimented on the *'Masked Face Recognition'* (Singh 2023a), *'Mask Detection and Masked Face Recognition'* (Ullah *et al.* 2022) and the *'Custom Real-Time Masked Face Recognition'* datasets. In the study, the *'Masked Face Recognition'* dataset by Singh (2023a), *'Mask Detection and Masked Face Recognition'* dataset by Ullah *et al.* (2022) and the *'Custom Real-Time Masked Face Recognition'* dataset are respectively labelled as *'MFR Dataset'*, *'MDMFR Dataset'* and *'CRMFR Dataset'*. The analysis and results begin with the dataset visualisation and summary, next the model results derived from the training phase are depicted and described in detail. In the last stage, model performance is discussed, highlighting evaluation metrics applied to determine the effectiveness of the proposed model in comparison to the existing research literature results. The section concludes with a summary based on the MFR model documented outcomes.

5.3.1. Mask Facial Recognition Datasets

In the context of the *'MFR Dataset'*, *'MDMFR Dataset'* and the *'CRMFR Dataset'*, this study explores MFR, presenting a comprehensive experimentation and the ensuing results based on the respective datasets. All three datasets contain multiple variations of images, including blurry, HD, various light intensities, diverse facial angles, and several images with cropped regions based on the applied augmentation techniques. All these images are utilised to aid the development of a generalised and robust model.

5.3.1.1. The 'MFR Dataset'

In the '*MFR Dataset'*, a total of 456 images exist within the original dataset. Images of the participants are compartmentalised into 19 folders as displayed in *Figure 5. 10*. Each folder consists of 24 images per participant. To ensure the model is robust and works optimally, data augmentation is created and applied to the original dataset as indicated in chapter four within the MFR preprocessing step under the *CNN seven-step model development.* Since there are 21 augmented images generated per folder, the updated dataset consists of $((24 \times 21) + 24) \times 19 = 10032$ images. Therefore, the dataset contains a balanced and equally distributed number of images with a total of 528 images per class.

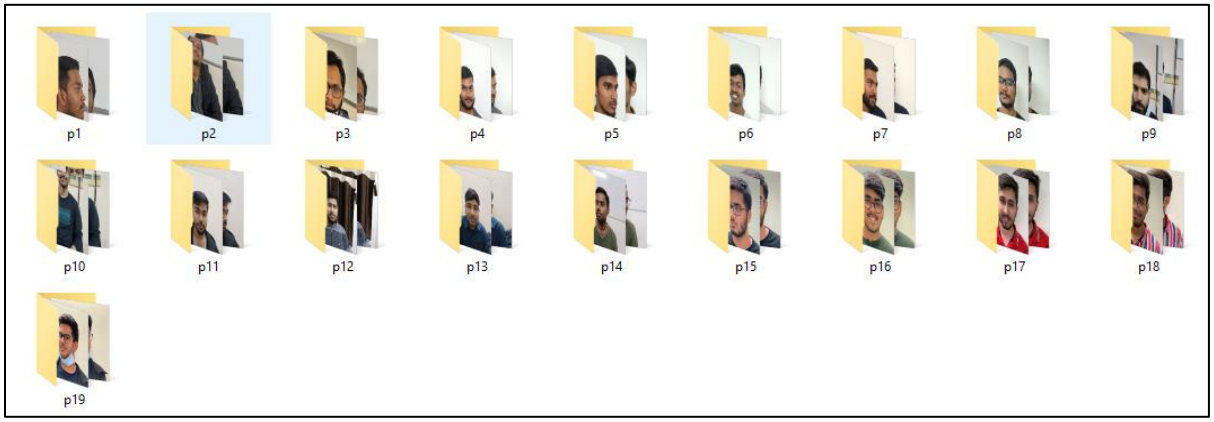


Figure 5. 10: Folders for Each Participant in the 'MFR Dataset' (Singh 2023a)

5.3.1.2. The 'MDMFR Dataset'

In the '*MDMFR Dataset'*, a total of 500 images exist within the original dataset. Images of the participants are compartmentalised into 50 folders as displayed in *Figure 5. 11*. Each folder consists of 10 images per participant. As applied to in the first dataset, the same data augmentation techniques are applied to the second dataset, creating a new batch of images. The updated dataset

therefore consists of $((10 \times 21) + 10) \times 50 = 11000$ images and contains a balanced and equally distributed number of images with a total of 220 images per folder.

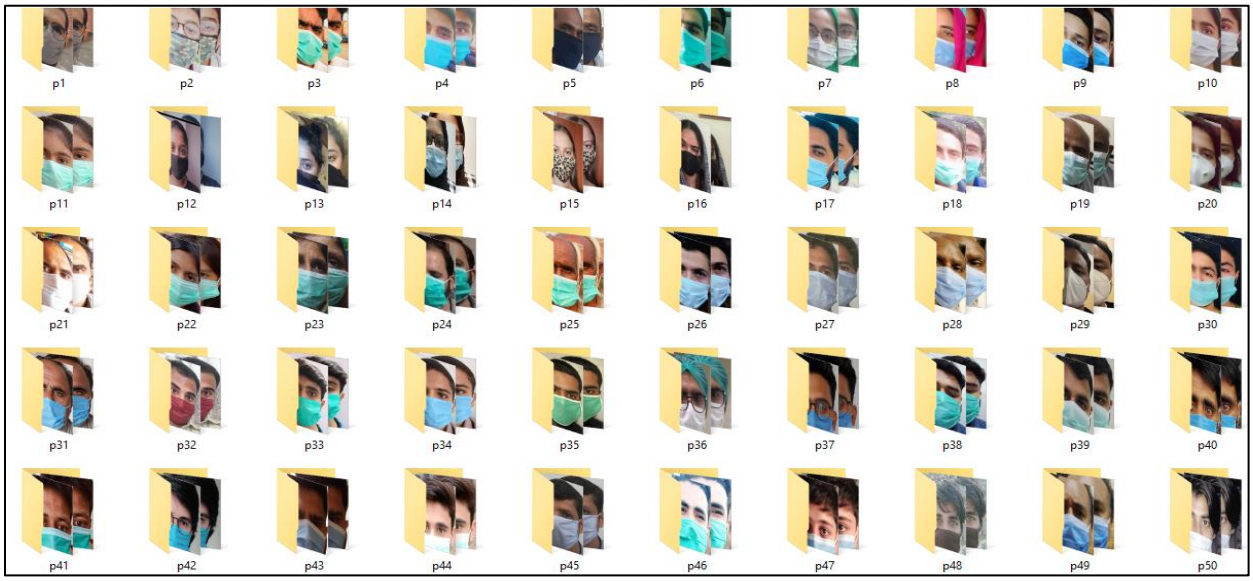


Figure 5. 11: Folders for Each Participant in the 'MDMFR Dataset' (Ullah et al. 2022)

5.3.1.3. The 'CRMFR Dataset'

Within the '*CRMFR Dataset'*, a total of 250 images exist within the original dataset. Images of the participants are compartmentalised into 10 folders as displayed in *Figure 5. 12*. Each folder consists of 25 images per participant. As applied to the first and second dataset, the same data augmentation techniques are applied to the third dataset to create a new batch of images. Therefore, the updated dataset consists of $((25 \times 21) + 25) \times 10 = 5500$ images and contains a balanced and equally distributed number of images with a total of 550 images per folder.

Figure 5. 12: Folders for Each Participant in the 'CRMFR Dataset'

Investigating the outcomes of training the MFR models provide critical insight into their performance and efficacy. The next section explores the results obtained from training the MFR model using the aforementioned MFR datasets in identifying the accurate identity of a masked face individual despite the model being presented with facial occlusions.

5.3.2. Masked Facial Recognition Model Training Results

The dataset images are trained using the CNN architecture, optimising parameters to enhance the accuracy in predicting or classifying new, unseen data to perform classification proficiently.

5.3.2.1. The 'MFR Dataset'

The model follows a dataset split of 80/10/10. This indicates that 80% of the data is employed for training the model, which consists of 8024 images. Within this split, 10% of the data, comprising 1004 images, is used for data validation during hyperparameter tuning, while the other 10%, also consisting of 1004 images, is reserved for testing the model on unseen test data. During the epoch iteration training, validation and testing process, the model's accuracy in the MFR phase is displayed as seen in *Annexure C, Figure 3.* The model accuracy is defined as the mean accuracy across individual batches of training data. The metric "val_accuracy" is precisely reflective of the accuracy based on the predictions within the validation data batch. The "val_loss" exhibited in the model diagram for each epoch quantifies the instances where predictions did not align with the values in the validation data batches. This loss metric reveals discrepancies identified within the model across various batches of data. The final epoch yields a validation accuracy of 0.9930 and a validation loss of 0.0380.

Once the model's epoch iteration process has been completed, it is effectively evaluated based on the performance of the deep learning model within the test portion of the unseen dataset. This evaluation is performed to calculate the average loss and accuracy over the entire test dataset. The model therefore produced an overall accuracy rate of 99.40%, highlighting the model's remarkable performance. The model in conjunction with the exceptional accuracy rate also produced an overall loss of 0.0305, indicating minimal discrepancies within the performance.

Upon completion of the trained, validated and tested images, through the utilisation of plotted graphs, a demonstration based on the evolution of key metrics across the epochs during the training process is visualised aptly.

The training, validation and testing accuracy, and training, validation and testing loss are illustrated in *Figure 5. 13*. Based on the training, validation and testing results obtained during the epoch iteration, there is a notable improvement in model performance over the epoch iteration. The training results based on the model's accuracy improved from 74.44% to a remarkable 100% in the final epoch and 91.53% to 99.30% in the validation accuracy. The test data, performed on the unseen test images, provided a test accuracy of 99.40%. The training loss significantly decreased from 1.2146 to 0.0142 whilst the validation loss followed a decrease from 2.6375 to 0.0380 in the last epoch. The test loss produced an outcome of 0.0305. This therefore highlights the model's ability to generalise on new and unseen data.

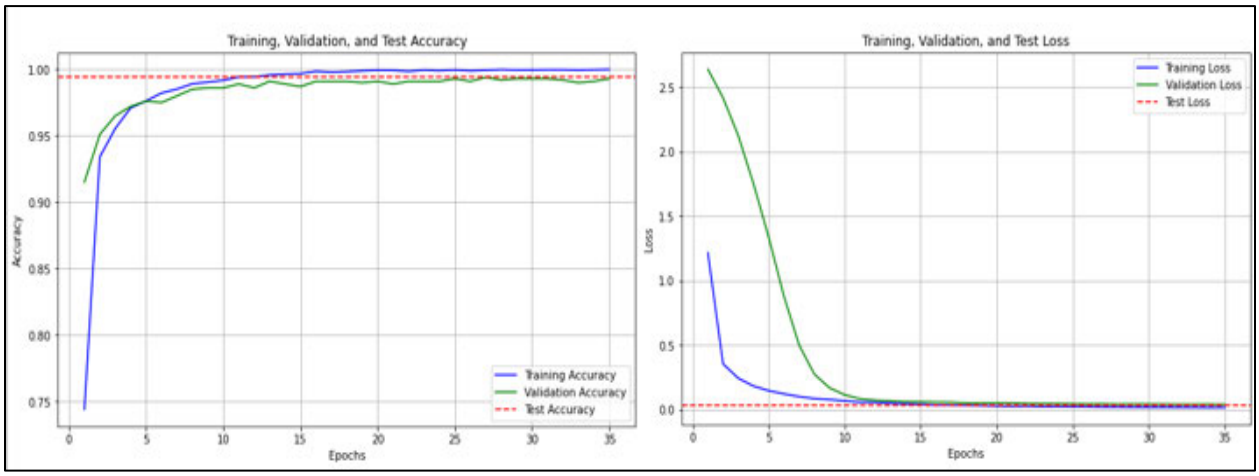


Figure 5. 13: Plotting the Epochs of the 'MFR Dataset'

Therefore, the consistent increasing trend in both the training and validation accuracy and decreasing trend in both the training and validation losses indicates no signs of overfitting. Due to the narrowing gap between training and validation accuracies, it further indicates a balanced fit. Thus, mitigating concerns of both overfitting and underfitting within the data.

5.3.2.2. The 'MDMFR Dataset'

Following the aforementioned dataset split, 8800 images are employed to train the model, whilst 1100 images are allocated for hyperparameter tuning during validation and 1100 images are reserved for testing the model on the unseen test data. The accuracy of the model's results during the training and validation epoch iteration process with the inclusion of the test accuracy and loss values are depicted in *Annexure C, Figure 4.* The last epoch displayed a validation accuracy of 0.9982 and a validation loss of 0.0259.

Once the model's epoch iteration process is completed, the model is evaluated based on its performance on the unseen test dataset. The model produced an overall accuracy rate of 99.82%, emphasising the model's outstanding performance. The model in conjunction with the exceptional accuracy rate also produced an overall loss of 0.0307, demonstrating insignificant discrepancies in the performance.

The training, validation and testing accuracy, and training, validation and testing loss are illustrated in *Figure 5. 14.* Within the context of the training and validation accuracy, the model displays a steady increase from 62.76% to 99.99% and 93.82% to 99.82%. Within the test data, performed on the unseen images, the model produced a test accuracy of 99.82%. In the training and validation loss the model displays a steady decrease from 2.0658 to 0.0203 and 3.6029 to 0.0259. The test loss produced an outcome of 0.0307. Therefore, this highlights the model's ability to generalise on unforeseen data and indicates the model's ability to capture complex patterns without overfitting.

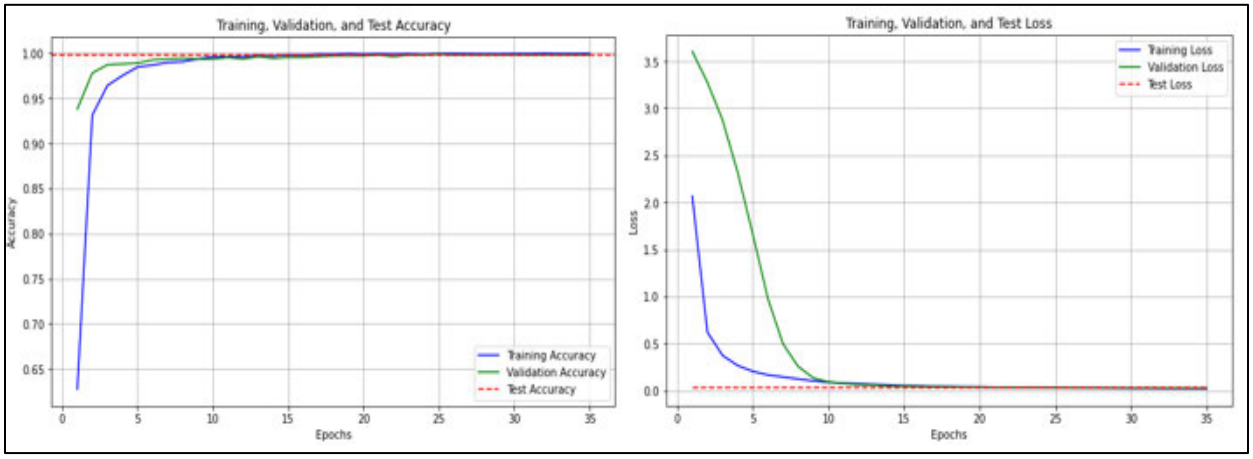


Figure 5. 14: Plotting the Epochs of the 'MDMFR Dataset'

Since there is a consistent increase in accuracy and decrease in the loss on both training and validation sets throughout the epochs, it suggests that no overfitting has occurred.

5.3.2.3. The 'CRMFR Dataset'

This dataset follows the same aforementioned split with 4400 images employed to train the model, whilst 550 images are allocated for hyperparameter tuning during validation and 550 images are reserved for testing the model on the unseen test data. The training, validation and testing epoch

iteration process for this dataset's model is depicted in *Annexure C, Figure 5*. The last epoch iteration displayed a validation accuracy of 0.9982 and a validation loss of 0.0097.

Upon completion of the model's epoch iteration process, the model is evaluated based on the performance on the unseen test dataset. The model produced an overall accuracy rate of 99.82%, emphasising the model's remarkable performance and an overall loss of 0.0094, indicating minimal discrepancies in performance.

The training, validation and testing accuracy, and training, validation and testing loss are illustrated in *Figure 5. 15.* In this dataset, the model begins with a training loss of 0.8622 and accuracy of 79.98%, converging to a loss of 0.0051 and accuracy of 100% in the concluding training epochs. In the validation accuracy and loss, a significant improvement is displayed, highlighting a decreasing loss from 2.0724 to 0.0097 and accuracy increasing from 94.55% to 99.82%. Within the test data, performed on the unseen images, the model produced a test accuracy of 99.82% and a test loss of 0.0094. These results portray the effectiveness of the training process and the robustness of the model in accurately classifying new unseen data presented to the model.

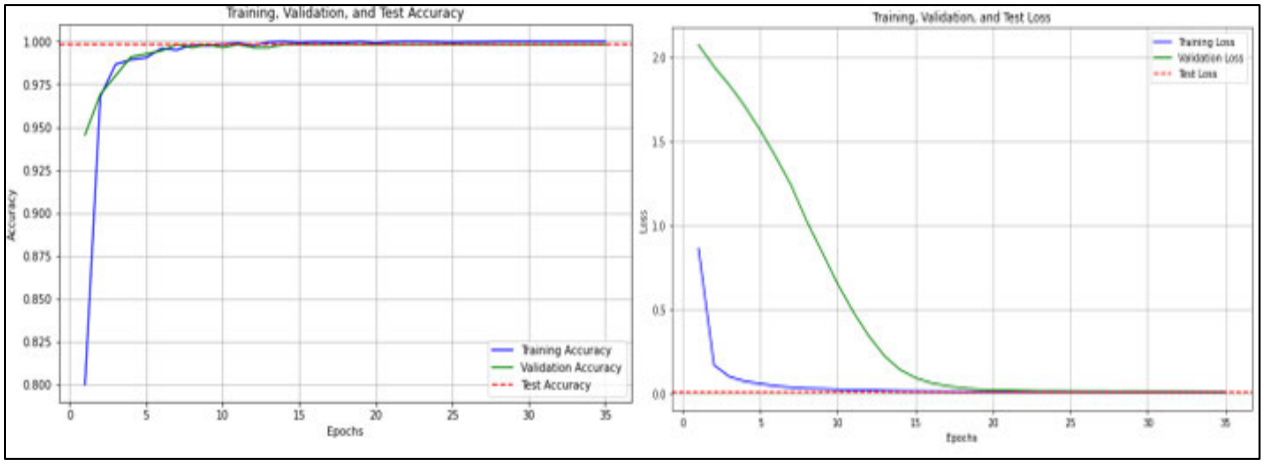


Figure 5. 15: Plotting the Epochs of the 'CRMFR Dataset'

Therefore, the minimal difference between the training and validation losses, along with the increasing trend in training and validation accuracies, indicates effective generalisation, with no signs of overfitting, affirming the model's robustness and ability to generalise on unseen data.

The evaluation of model performance through the confusion matrix and evaluation metrics offers valuable insights into its effectiveness and robustness. The results of these metrics based on model performance are illustrated both quantitatively and qualitatively within the *model performance* section.

5.3.3. Masked Facial Recognition Model Performance

The performance of the model is evaluated on the *'MFR Dataset'*, *'MDMFR Dataset'* and the *'CRMFR Dataset'* using the confusion matrix, accuracy, precision, recall, and the f1-score.

5.3.3.1. Model Performance – 'MFR Dataset'

Visual insight into the accuracy of all participants within the dataset during the testing phase is depicted in *Figure 5. 16*, utilising a confusion matrix. In the aforementioned detailed description based on the dataset split, the dataset underwent a division into training, validation and testing sets, with an 80/10/10 split. It is noteworthy that a total of 1004 images are dedicated to testing the model, with an allocation of 53 images per participant in each class during the testing phase. The diagonal elements, colour coded in a dark shade of blue represent the number of correct predictions for each class. Conversely, the off-diagonal elements represent the number of misclassifications between the relevant classes. All classes contain a respectable number of correct classifications whilst the minimum number of correct classifications are 51, that is obtained for participant p8. This indicates that only two misclassifications are yielded by the model for participant p8. Therefore, the number of misclassifications produced by the model for each participant fall within the range of zero to two. Hence, this suggests a slight discrepancy between the classifications and misclassifications, highlighting the model's effectiveness in accurately predicting the correct identity of the participants for each class.

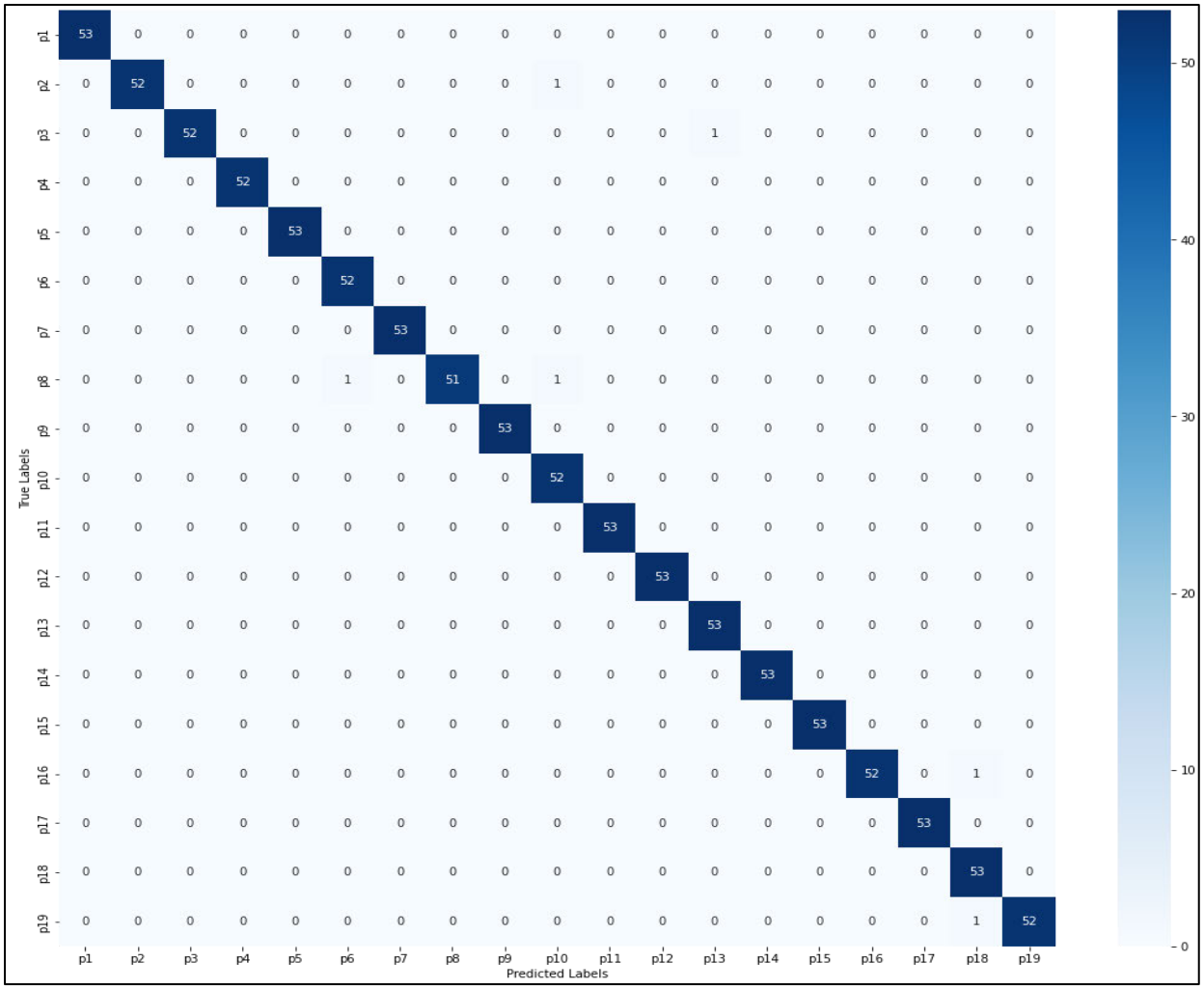


Figure 5. 16: Confusion Matrix on the 'MFR Dataset'

By adding all the *true positives* (998) for each class and dividing it by the total number of testing images (1004), it provides insight into whether the model is overfitting or not. Performing this calculation results in a total of 99.40%. Hence, there is no overfitting since the unseen test data in the testing result is 99.40% and the confusion matrix calculation is also 99.40%.

Within the context of performing a model comparison using the *'MFR Dataset'*, which was released recently in 2023, it has therefore not been widely explored by other studies. Moreover, within the dataset, no other studies in the existing literature utilised this dataset during the training and assessment of model performance as illustrated in *Table 5. 9*. Acquiring publicly available MFR datasets proved challenging within this research study.

Table 5. 9: Performance Evaluation on the 'MFR Dataset'

Study	Accuracy	Precision	Recall	F1-score
Proposed Study	99.40	99.41	99.40	99.40

The study however, yielded a respectable result of 99.40% across the accuracy, recall and f1-score while the precision attained a score of 99.41%.

Using a set of sample images from a combination of the *'MFR Dataset'*, *'MDMFR Dataset'* and the *'CRMFR Dataset'* allocated test data, the model performance is further assessed. The predicted results based on the set of test images are illustrated and described aptly in *Table 5. 10* and *Figure 5. 17*.

Table 5. 10: Results of the 'MFR Dataset' Sample Prediction

Label	Description	Accuracy Percentage (%)	Explanation
A	P16 with mask on.	99.94	This indicates that the model is 99.94% certain that this is participant p16.
B	P15 with mask on.	100	This indicates that the model is 100% certain that this is participant p15.
C	Unknown prediction with mask on.	97.82	A test dataset image taken at random from the *'CRMFR Dataset'* is tested on the model. The model was able to accurately identify that this participant did not belong to the '*MFR Dataset'*.
D	Uknown prediction with mask on, at an angle and in the dark.	96.95	A test image taken at random from the '*MDMFR Dataset'* with darkening applied through the implementation of augmentation techniques was tested on. The model was able to accurately identify that this participant did not belong to the *'MFR Dataset'*.
E	P5 prediction with mask on,	99.94	Despite significant image darkening to simulate varying lighting conditions using augmentation techniques and presenting the

	at an angle and in the dark.		model with an image of an angled face, it was still able to accurately identify the participant p5 and achieved a respectable accuracy rate.
F	P17 prediction with mask on, at an angle and in the dark.	99.58	Despite even further significant image darkening to simulate varying lighting conditions using augmentation techniques and presenting the model with an image of an angled face, it was still able to accurately identify the participant p17 and achieved a respectable accuracy rate.
G	P18 prediction with mask on and at an angle.	99.89	Despite applying different colour variations to the image to simulate varying lighting conditions using augmentation techniques and presenting the model with an image of an extreme right-angled face, it was still able to accurately identify the participant p18 and achieved a respectable accuracy rate.
H	P1 prediction with mask on, at an angle and slight darkening to the image.	99.62	Despite even further significant image darkening and applying different colour variations to the image to simulate varying lighting conditions using augmentation techniques and presenting the model with an image of an extreme right-angled face, it was still able to accurately identify the participant p1 and achieved a respectable accuracy rate.

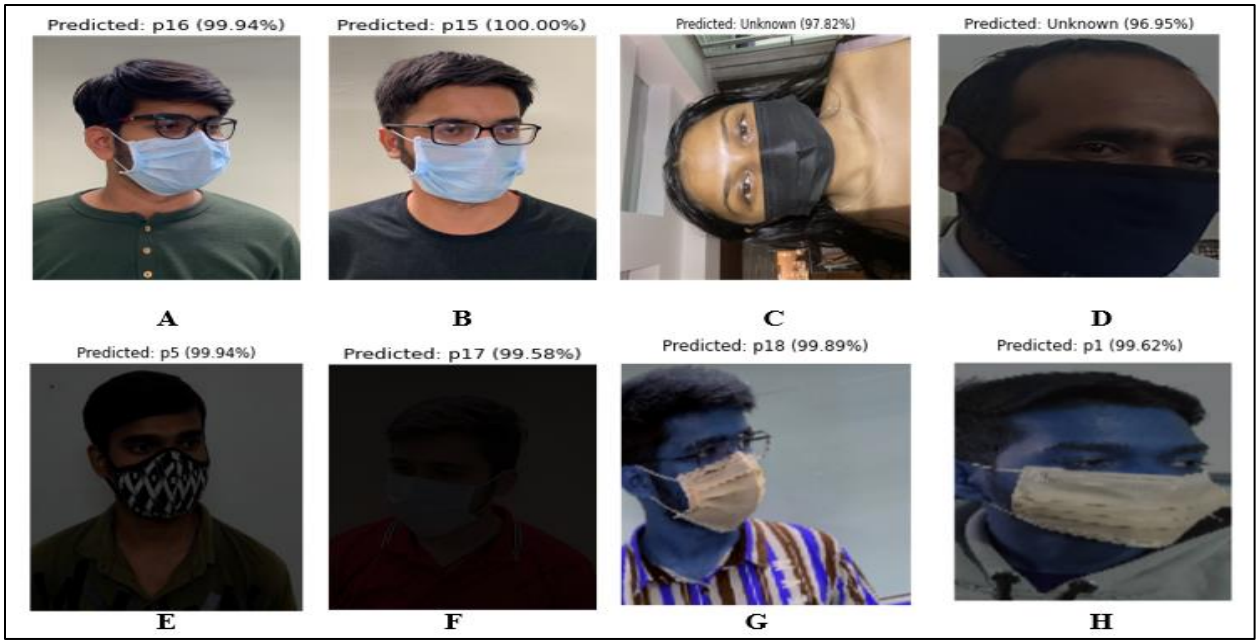


Figure 5. 17: 'MFR Dataset' Sample Prediction (Ullah et al. 2022; Singh 2023a)

The model attained remarkable accuracy scores within the range of 96.95% to 100%. Despite recording the lowest accuracy score of 95.96%, the model was still certain that the *'MDMFR Dataset'* participant did not belong to this dataset, despite being presented with image darkening, and the image presented at an angle. Furthermore, the model possessed the ability to recognise participant p16, p15 and p18 in label *A*, *B and G* despite their glasses covering a region of their face and achieved an accuracy score of 100% for label *B*.

5.3.3.2. Model Performance – 'MDMFR' Dataset

The confusion matrix depicted in *Figure 5. 18,* illustrates all participants within the dataset. It is noteworthy that a total of 1100 images are dedicated to testing the data, with an allocation of 22 images per participant in each class. In the classification process, all classes contain a respectable number of precise and accurate classifications whilst the minimum number of correct classifications is 21, which are obtained for participants p29 and p37. Therefore, the number of misclassifications produced by the model for each participant fall within the range of zero to one. Hence, this suggests a slight discrepancy between the classifications and misclassifications, highlighting the model's effectiveness in accurately predicting the correct identity of the participants for each class.

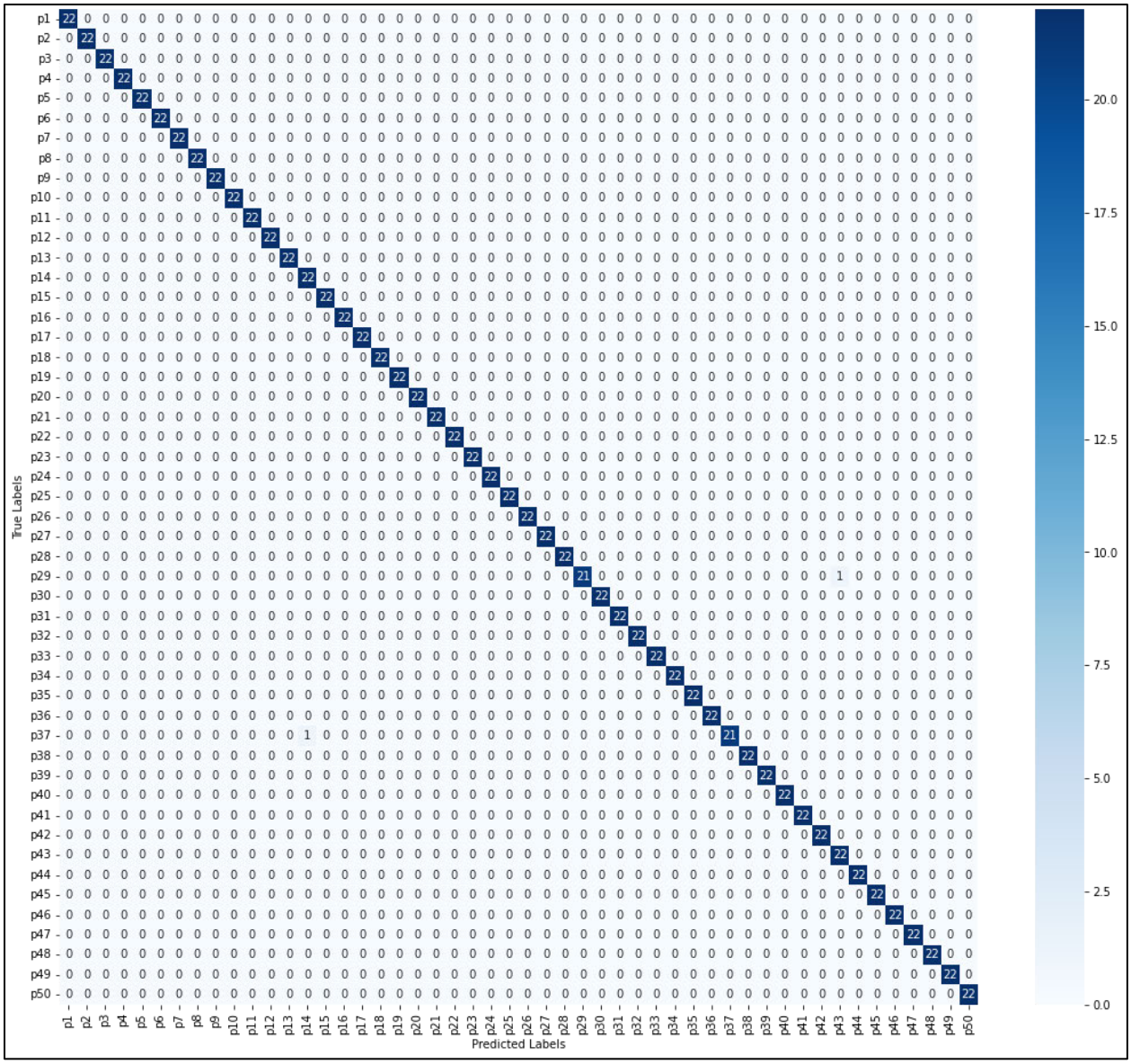


Figure 5. 18: Confusion Matrix on the 'MDMFR Dataset'

By adding all the *true positives* (1098) for each class and then expressing it as a fraction over the total number of test images (1100), it provides insight into whether the model is overfitting or not. Performing this calculation results in a total of 99.82%. Hence, there is no overfitting since the unseen test data in the testing result is 99.82% and the confusion matrix calculation is also 99.82%.

A comparison of the model is performed against the existing literature based on the evaluation metrics that include the accuracy, precision, recall, and f1-score. A quantitative comparison is therefore performed between the proposed model and existing literature model that use the same

'MDMFR Dataset' for the purpose of training and testing the models, as depicted in *Table 5. 11*. Referring to *Annexure A*, the entire study result for the model can be found based on the corresponding author name. Furthermore, it is important to note that the *'MDMFR Dataset'*, developed by Ullah et al. (2022), is a unique dataset. Therefore, it has not been extensively explored by other studies for training and testing models, thus it has only been specifically identified and highlighted in the existing review article.

Table 5. 11: Performance Evaluation on the 'MDMFR Dataset'

Study	**Accuracy**	**Precision**	**Recall**	**F1-score**
Proposed Study	99.82	99.82	99.82	99.82
(Ullah *et al.* 2022)	93.33	93.00	94.50	93.74

Based on the comparison of evaluation metrics within *Table 5. 11*, it is evident that the proposed model outperformed the model developed by author Ullah *et al.* (2022) when utilising the same dataset. In performing this analysis, the model displayed an outstanding performance within the accuracy, precision, recall and f1-score, achieving a 99.82% score across the relevant metrics. Furthermore, it is important to note the identified limitations within the existing study have been effectively addressed within the proposed model. The MFR model developed by Ullah *et al.* (2022), demonstrated limitations, with accuracy, precision, recall, and f1-score all falling below the 95% threshold. This highlighted the need for further refinement and exploration of an alternative model or technique to enhance performance. Additionally, the computational complexity associated with the training process warrants consideration, prompting a balance between model sophistication and computational efficiency in future endeavours. Lastly the study noted that it would like to implement testing on a more diverse and quantified dataset, indicating that there is a lack of diversity within the FMD and MFR datasets. Hence, the model requires significant and further improvement. Therefore, the proposed MFR model attained exceptional evaluation metric scores when compared to the existing literature identified for this dataset. This highlights the proposed model's ability to discover patterns and conduct accurate classifications on the respective images in this dataset.

Using *Figure 5. 19,* a graph-oriented visualisation of the proposed model is depicted. The graph consists of a comparison between the proposed model developed and the research model developed by author Ullah *et al.* (2022), utilising the same *'MDMFR Dataset.'*

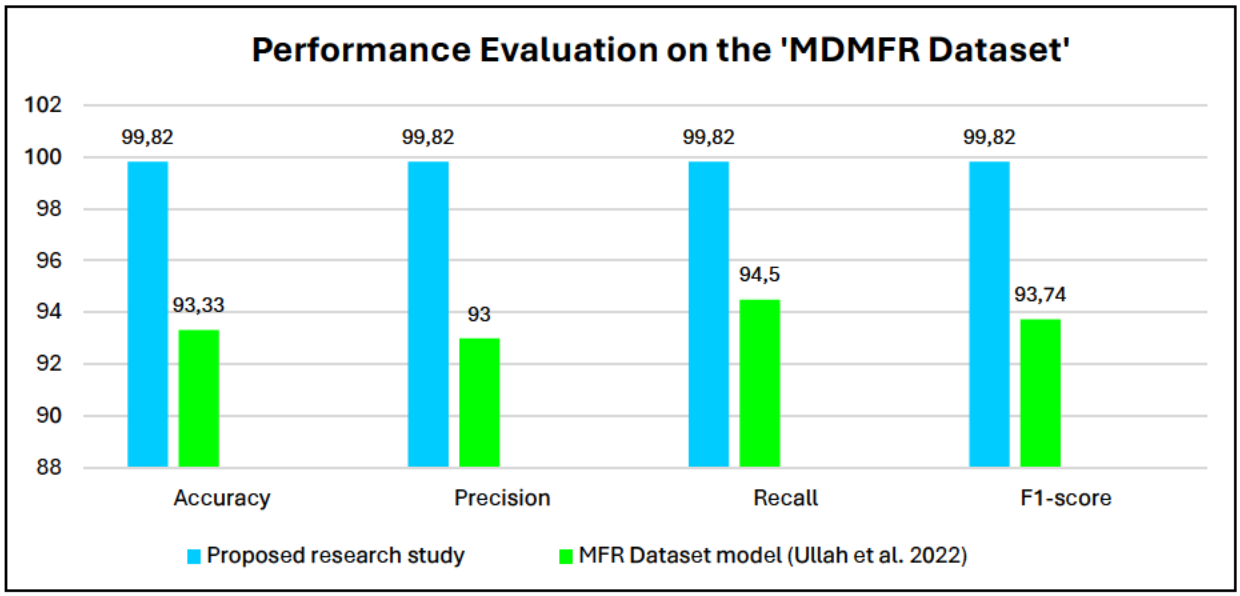


Figure 5. 19: Performance Evaluation on the 'MDMFR Dataset' Graph

Using sample images from the *'MFR Dataset'*, *'MDMFR Dataset'*, *'CRMFR Dataset'*, allocated test data and the *'face mask detection'* dataset, the model performance is further assessed. The predicted results based on the set of test images are illustrated and described in *Table 5. 12 and Figure 5. 20.*

Table 5. 12: Results of the 'MDMFR Dataset' Sample Prediction

Label	Description	Accuracy Percentage (%)	Explanation
A	P17 with mask on.	100	This indicates that the model is 100% certain that this is participant p17 despite being presented with a slight facial angle.
B	P7 with mask on at an angle.	99.99	This indicates that the model is 99.99% certain that this is participant p7.
C	Unknown prediction with	95.32	A test image taken at random from the *'MFR Dataset'* is tested on the model. The model was able to

	mask on at an angle.		accurately identify that this participant did not belong to the *'MDMFR Dataset'* despite being presented with a facial angle.
D	Unknown prediction with mask on.	99.97	A test image taken at random from the *'face mask detection'* dataset was tested on the model. The model was able to accurately identify that this participant did not belong to the *'MDMFR Dataset'*.
E	P13 prediction with mask on, at an angle and in the dark.	99.64	Despite introducing significant image darkening to simulate varying lighting conditions and presenting the model with an image containing an angled face, the model was still able to accurately identify participant p13 and produced a respectable accuracy rate.
F	P5 prediction with mask on, at an angle and in the dark.	100	Despite introducing significant image darkening to simulate varying lighting conditions and presenting the model with an image containing an angled face, the model was still able to accurately identify participant p5 and produced a respectable accuracy rate.
G	P30 prediction with mask on and slight darkening to the image.	99.90	Despite even further significant image darkening and applying different colour variations to the image to simulate varying lighting conditions using augmentation techniques, it was still able to accurately identify the participant p30 and achieved a respectable accuracy rate.
H	P8 prediction with mask on and head at an angle.	99.80	Despite applying different colour variations to the image to simulate varying lighting conditions using augmentation techniques and presenting the model with an image of a slight right-angled face, it was still able to accurately identify the participant p8 and achieved a respectable accuracy rate.

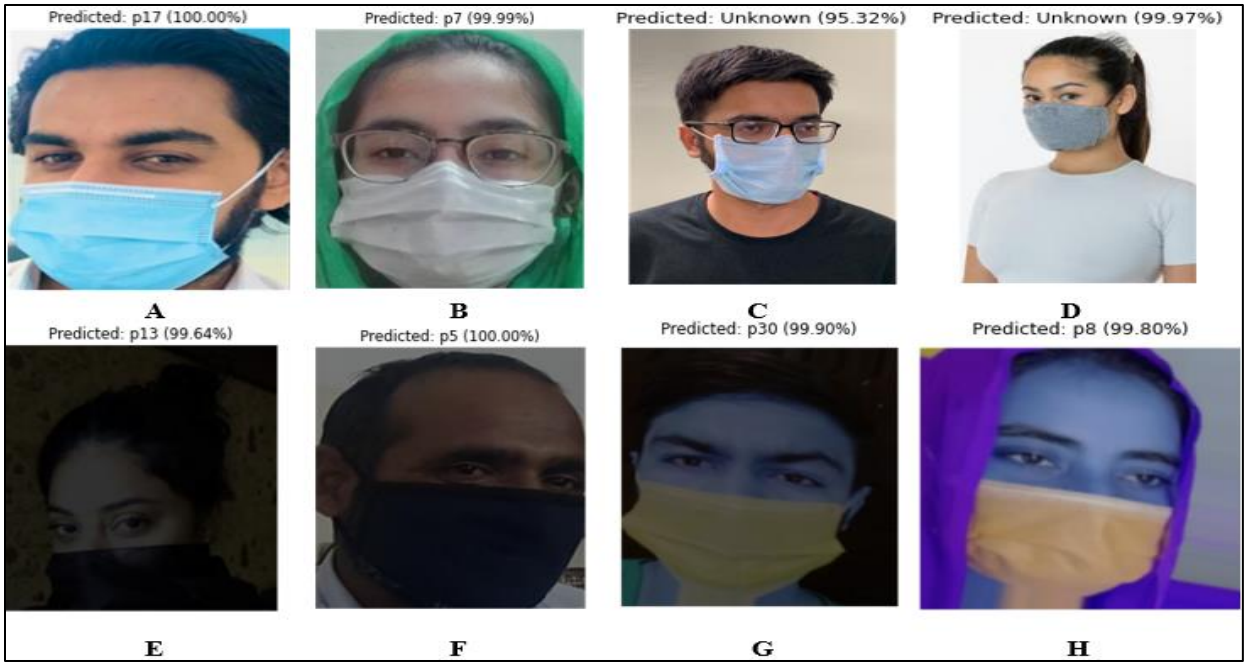


Figure 5. 20: 'MDMFR Dataset' Sample Prediction (Gurav 2020; Ullah et al. 2022; Singh 2023a)

The model achieved respectable accuracy scores within the range of 95.32% to 100%. Despite recording the lowest accuracy score of 95.32%, the model confidently determined that the *'MFR Dataset'* participant did not belong to the dataset, even when the participant was wearing glasses and presented an angled facial orientation. Furthermore, it demonstrates the model's ability to recognise participants with objects covering regions of their faces, such as headscarves and glasses as illustrated in label *B, C* and *H*, where the model produced an accuracy score of 99.99% for participant p7.

5.3.3.3. Model Performance – 'CRMFR' Dataset

The confusion matrix depicted in *Figure 5. 21,* illustrates all participants that exist within this dataset. It is noteworthy that a total of 550 images are dedicated to testing the data, with an allocation of 55 images per participant in each class during the testing phase. In the classification process, all classes contain a respectable number of correct classifications whilst only one class presented a single misclassification with a value of 54 out 55 classifications. This misclassification is obtained for the participant p9. Hence, this suggests a slight discrepancy between the classifications and misclassifications, highlighting the model's effectiveness in accurately predicting the correct identity of the participants for each class.

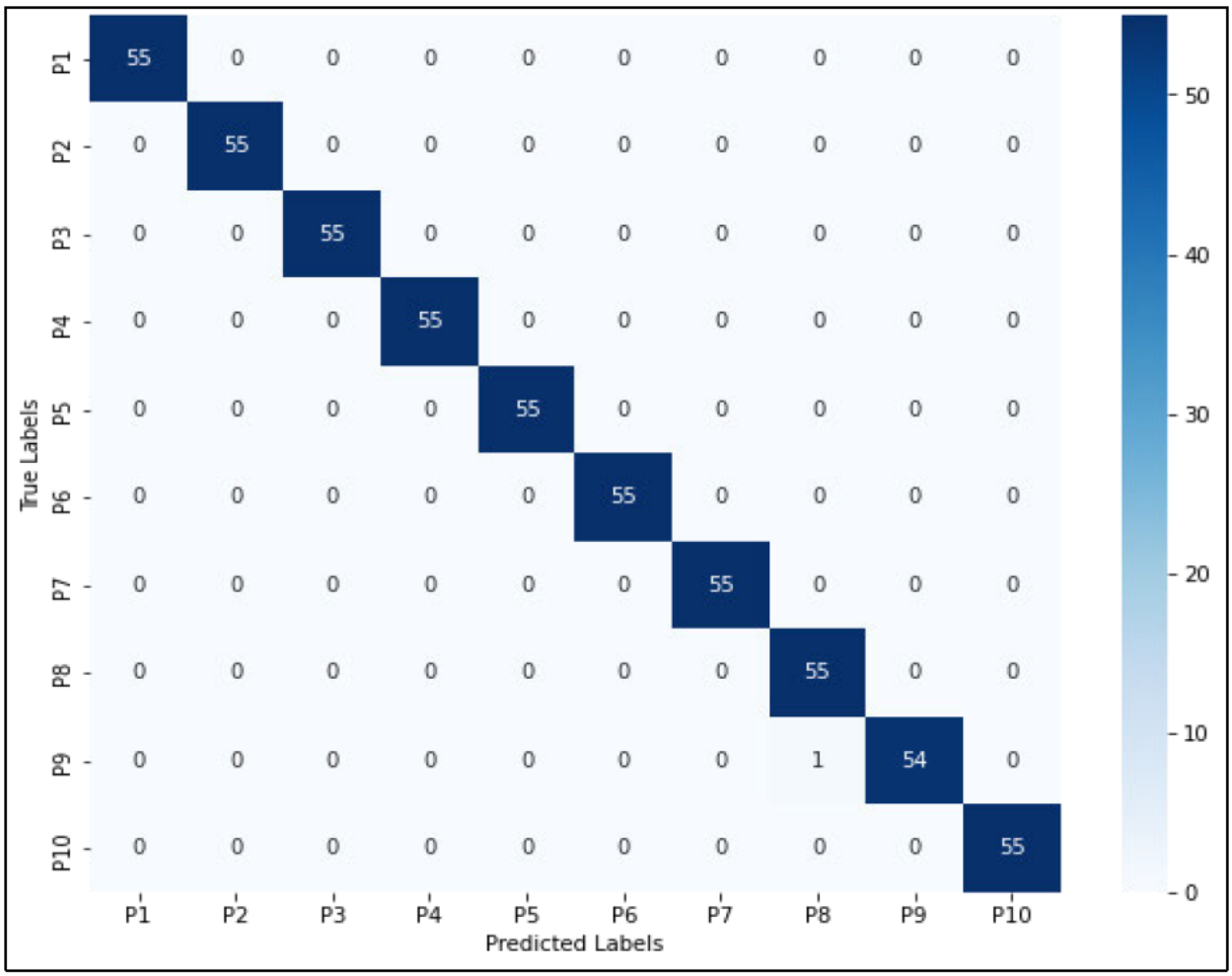


Figure 5. 21: Confusion Matrix on the 'CRMFR Dataset'

By adding all the *true positives* (549) for each class and dividing it by the total number of testing images (550), it provides insight into whether the model is overfitting or not. Performing this calculation results in a total of 99.82%. Hence, there is no overfitting since the unseen test data in the testing result is 99.82% and the confusion matrix calculation is also 99.82%.

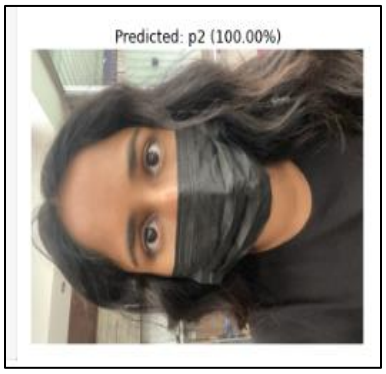
Since the *'CRMFR Dataset'* is a custom dataset, there are no comparisons to be made in the existing literature as depicted in *Table 5. 13*. The study however produced an outstanding accuracy, precision, recall and f1-score of 99.82% across the metrics.

Table 5. 13: Performance Evaluation on the 'CRMFR Dataset'

Study	**Accuracy**	**Precision**	**Recall**	**F1-score**
Proposed Study	99.82	99.82	99.82	99.82

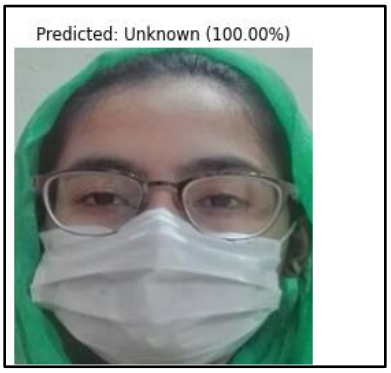
The series of images below present an illustration based on simulated and real-time recognition for MFR. In *labels A to W*, the anticipated result is evaluated on participants allocated test images both within and outside the *'CRMFR Dataset'*. A confidence threshold of 0.8 (i.e., 80%) was set. In *A,* the first image presents a confidence score of 100% indicating the model is 100% certain that it was participant p2. In *B*, the model is 100% confident that it was participant p7. In labels *C* and *D*, random allocated test images were taken from the *'MDMFR Dataset'* and the *'MFR Dataset'* and were then tested on the *'CRMFR Dataset'* model. The model was able to accurately identify that these participants did not belong to this dataset. The model, therefore, displayed a confidence score of 100% that the participants did not belong to the dataset for both the images.

Since a real-time dataset exists with images of participants, real-time recognition can be performed. The series of images based on labels *E to W,* illustrate real-time recognition on four participants (i.e., two that exist within the dataset and two that do not belong to the dataset). In performing real-time recognition, the following results were achieved. Labels *E and F* illustrate participants that did not belong to the dataset that were tested within the dataset in real-time. The model was able to accurately identify that these participants did not belong to this dataset. In the real-time scenario the model is set to provide a "*0.00"* accuracy rate when the model predicts a participant that is below the threshold value set (i.e., 0.8) and does not belong to the specified dataset when performing real-time recognition. In labels *G to K* and *P to S*, real-time recognition was conducted using a normal standard home lighting condition of 50 lux and utilised surgical face masks. Extreme variation in angles is presented on the frontal face, and on both the left and right-hand side as seen in labels *H, I, J , K , Q, R and S*. As seen in *K,* the model was even able to accurately recognise participant p2 whilst their eyes are almost completely closed. The model was also able to recognise a face with or without the face mask on as seen on participant p2 in *G* and *H* and then in *P* and *Q* on participant p7. In *L to O* and *T to W,* the same experiments conducted with the normal lighting conditions are conducted with the lights completely switched off on both participant p2 and p7. The real-time recognition was conducted in extreme dimmed lighting conditions and utilised surgical face masks. In addition, the model was presented with extreme variations in facial angles (i.e., frontal face, left and right-hand side) as seen in labels *M*, *N, O, U, V* and *W*. The model performed optimally with the lights switched off and was able to recognise the relevant participant aptly and accurately with their face mask on and off as seen in *labels L, M, T and U*.

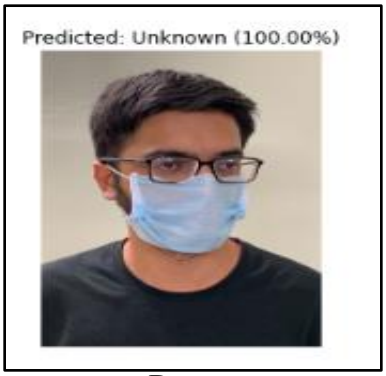


A

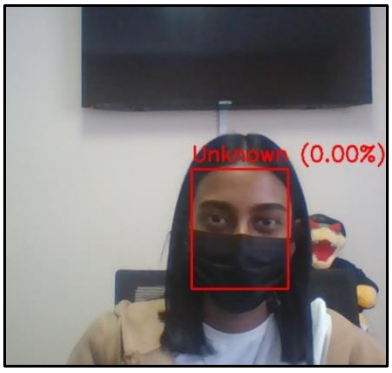
B



C

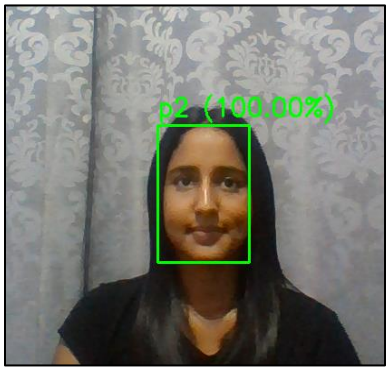


D

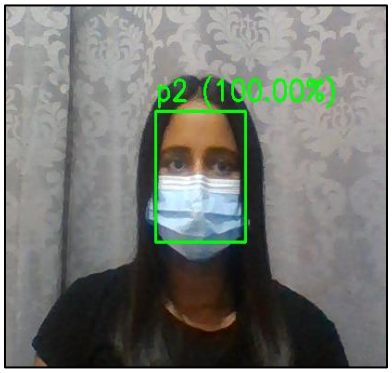


E

F



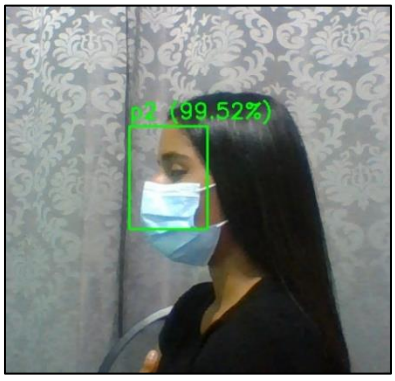
G



H

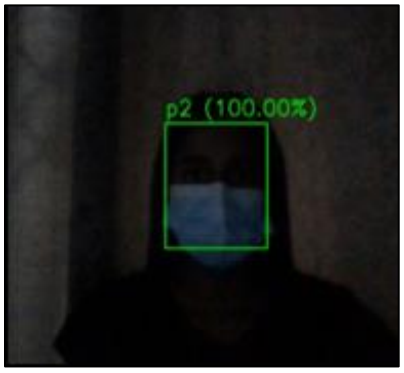
I

J



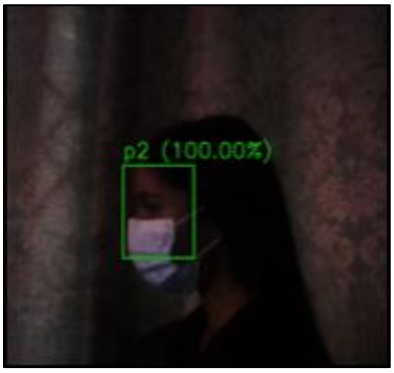
K

L

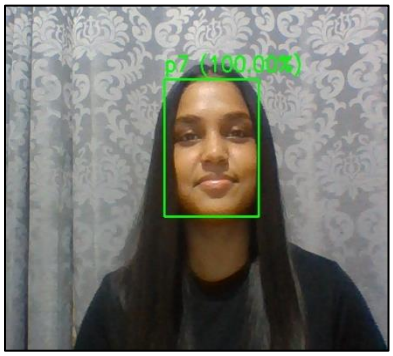


M

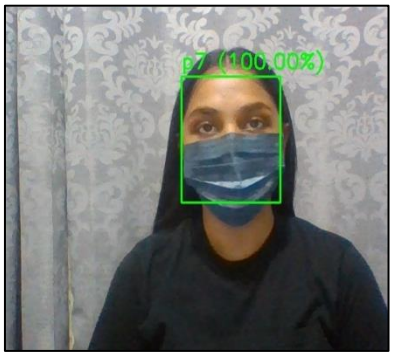
N



O

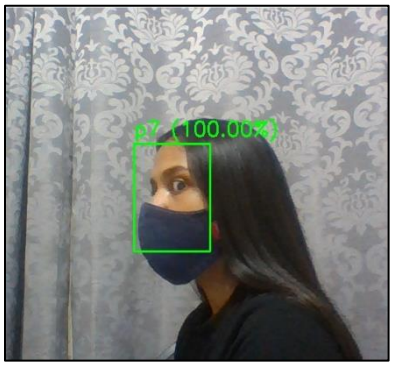


P



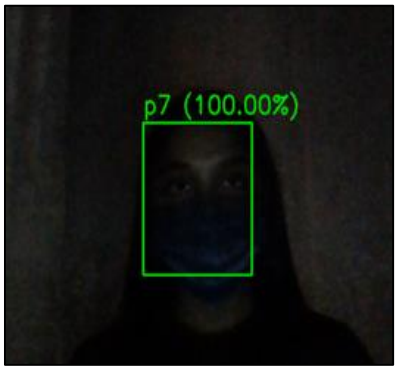
Q

R



S

T



U

V

W

Table 5. 14 provides a detailed description based on the model's predictions utilising the *'CRMFR Dataset'*.

Table 5. 14: Sample Image and Real-Time Recognition Results on the 'CRMFR Dataset'

Label	Description	Accuracy Percentage (%)
A	Participant p2 with mask on.	100
B	Participant p7 with mask on.	100
C	Unknown prediction with mask on.	100
D	Unknown prediction with mask on.	100
E	Real-time unknown prediction with mask on.	0.00
F	Real-time unknown prediction with mask on.	0.00
G	No mask, frontal face, accurate recognition on participant p2.	100
H	Mask on, frontal face, accurate recognition on participant p2.	100
I	Mask on, left angle, accurate recognition on participant p2.	100
J	Mask on, right angle, accurate recognition on participant p2.	100
K	Mask on, left angle, eyes almost closed, accurate recognition on participant p2.	99.52
L	No mask, frontal face, accurate recognition in the dark on participant p2.	100
M	Mask on, frontal face, accurate recognition in the dark on participant p2.	100
N	Mask on, right angle, accurate recognition in the dark on participant p2.	100
O	Mask on, left angle, accurate recognition in the dark on. participant p2.	100
P	No mask, frontal face, accurate recognition on participant p7.	100
Q	Mask on, frontal face, accurate recognition on participant p7.	100
R	Mask on, right angle, accurate recognition on participant p7.	100
S	Mask on, left angle, accurate recognition on participant p7.	100
T	No mask, frontal face, accurate recognition in the dark on participant p7.	100
U	Mask on, frontal face, accurate recognition in the dark on participant p7.	100
V	Mask on, right angle, accurate recognition in the dark on participant p7.	100
W	Mask on, left angle, accurate recognition in the dark on participant p7.	100

During both testing on the sample test images and in real-time recognition, the model displayed exceptional results in its ability to detect and recognise masked face participants despite the aforementioned occlusions of the facial region. During the real-time recognition phase, the model produced accuracy results that ranged from a perfect 100% score to the minimum score of 99.52% obtained when participant p2's eyes were closed, and facial region was at an extreme left angle. The model however was still able to accurately identify participants despite extreme varied lighting conditions, occlusions within certain facial regions, distorted blurry images, diverse facial angles and object obstructions including eye glasses that cover regions of the nose and eyes, hair down slightly covering the facial structure and head scarves that cover regions of the face that were presented to the model.

5.3.4. Masked Facial Recognition Summary

Within in the current developed MFR model, the overall mean accuracy score achieved on the unseen test data utilising all three of the MFR datasets is calculated as $99.40 + 99.82 + 99.82 = \frac{299.04}{3} = 99.68\%$ in accuracy. In addition, the combined average loss achieved for the MFR model is calculated as $3.05 + 3.07 + 0.94 = \frac{7.06}{3} = 2.35$. The model achieved a total mean accuracy of 99.68% and a total mean loss of 2.35. The proposed model therefore outperforms the exiting models highlighted in the MFR portion of the meta-analysis depicted in *Annexure A*. In addition to the developed model displaying an outstanding performance whilst achieving exceptional accuracy rates and remarkably low overall loss scores, the model presented notable achievements. The model when run on each of the three datasets, took a total of one hour and 30 minutes each to run with low computational complexity given the laptop hardware described above and possessed the ability to recognise the identity of a masked face individual despite being presented with dimmed lighting conditions, diverse facial angles, occlusions and obstructions to certain facial regions, and distorted blurry images.

Furthermore, it is important to note that within the 11 studies that were developed for MFR presented in the existing literature, six studies (Talahua *et al.* 2021; Ullah *et al.* 2022; Chen *et al.* 2023; Kumar *et al.* 2023; Peng *et al.* 2023; Wang, Li and Zou 2023) utilised a custom dataset or a dataset that is not publicly available. The remaining five studies (Cimmino *et al.* 2022; Marwa and Kais 2022; Pann and Lee 2022; Akingbesote *et al.* 2023; Al-Dmour *et al.* 2023) were composed of significantly large datasets that contain a minimum of 42 000 images to a maximum of

2 024 897 images. Therefore, given the excessive volume of images throughout the five datasets, a more streamlined selection was deemed appropriate for the study, as the vast quantity exceeded the study's requirements.

5.4. Chapter Summary

In the final analysis, it is evident based on the results and discussion, that both the FMD and MFR model achieved extremely high accuracies and tremendously low loss values. Overall, the FMD model achieved an overall average accuracy of 100% whilst the MFR model achieved an average of 99.68%. The MFR model outperformed all the existing MFR models that were identified in the systematic literature review based on the meta-analysis that was performed in chapter three. The proposed model was able to collectively perform both FMD and MFR despite the challenge of adverse lighting conditions, blurry images, being presented with varied angles, fake mask scenarios, incorrect use of a face mask, closed eye scenarios, occlusions of the face, object obstructions and intensive computational complexity. This further re-enforces the ability of the pre-trained MobileNetV2 architecture and the CNN with FaceNet InceptionResNetV1 architecture which can be utilised with aid of hyperparameter tuning to attain respectable results for both FMD and MFR. The next chapter provides a conclusion based on the final remarks, recommendations, and future work within the context of the proposed study.

CHAPTER SIX: SUMMARY, CONCLUSION AND FUTURE WORK

In this chapter, the focus is shifted to the study's outcomes, offering recommendations, discussing the limitations, and shedding light on future prospects for implementing the developed Face Mask Detection (FMD) and Masked Facial Recognition (MFR) models.

6.1. Summary

The research investigation has been pivotal in developing a FMD and MFR model. These models aim to address the shortcomings identified within the current face recognition models caused by the COVID-19 pandemic and other respiratory illnesses, as noted by Desai and Mehrotra (2020). Furthermore, the developed models assisted in eliminating the current issues identified in the existing literature for the identification of facial masks and the recognition of masked faces. The successful execution of the models involved the adept application of computer vision methodologies, and the utilisation of tailored algorithms designed for the detection of face masks and occluded facial features, and the subsequent recognition of masked faces. The models were developed through the utilisation of a powerful deep learning algorithm known as the Convolutional Neural Network (CNN).

At the outset of the study, the following objectives were formulated and consolidated:

i. To critically analyse the existing literature using the Systematic Literature Review (SLR) based on the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) protocol to identify the current trends in FMD and MFR models.
ii. To develop a hybrid CNN based FMD and MFR model to classify and recognise masked faces.
iii. To evaluate the performance of the newly developed FMD and MFR models by comparing them against the existing models using well-known evaluation metrics.

Aligned with these objectives, each subsequent objective was successfully achieved and is outlined as follows:

Within the first objective, the meta-analysis was conducted to identify existing trends and patterns highlighted in the current FMD and MFR models. During the implementation of the meta-analysis

several limitations were identified. The first limitation highlighted that in the exploration of the meta-analysis, a total of 72 research studies were reviewed with a total of 61 studies on FMD, eight for MFR and three for both FMD and MFR combined. This emphasised the lack of research within the MFR models and the combined models for both FMD and MFR developed together. Additionally, a notable trend emerged with a significant portion of studies reporting evaluation metric scores of 95% and below, inclusive of accuracy. Moreover, challenges were evident in both FMD and MFR models concerning accurate mask detection and masked facial recognition, particularly under adverse lighting conditions and varying angles of the human face. Notably, these models faced difficulty in accurately detecting smaller faces and low-resolution targets, often overlooking them. In addition, a common limitation surfaced as the models struggled to differentiate situations where a person incorrectly wore a face mask versus not wearing one at all. Furthermore, many of the developed models struggled with significant computational complexity, requiring substantial hours to train the models. Hence, the identified limitations presented a need for the development of a FMD and MFR model highlighted within the second objective.

The second objective aimed to build a hybrid model for FMD and MFR. This objective was achieved through the utilisation of MobileNetV2, FaceNet InceptionResNetV1, and CNN for model development. In the FMD phase of the model, the pre-trained MobileNetV2 model was utilised with the aid of hyper parameter tuning to build a model capable of FMD. The MTCNN for face and landmark detection, pre-trained FaceNet InceptionResNetV1 for feature extraction and the Euclidean distance metric for numeric feature matching significantly contributed toward the MFR development phase of the research. Both models utilised a total of five datasets. The *'face mask detection'* (1) and *'Face Mask Detection 12K images'* (2) datasets were utilised for FMD whilst the *'Masked Face Recognition'* (3)*, 'Mask Detection and Masked Face Recognition'* (4) and the *'Custom Real-Time Masked Face Recognition'* (5) datasets were utilised for MFR. The model designed for FMD demonstrated a mean accuracy level of 100% and the model developed for MFR achieved a mean accuracy rate of 99.68%. A standard definition web camera measuring 640 pixels in width, 480 pixels in height, and having a resolution of 96 dpi was employed during model testing based on the aforementioned laptop utilised. The successful development of the FMD and MFR model, therefore, aided in the achievement of the third objective.

To achieve the third objective, a comprehensive comparison was conducted by employing well-established evaluation metrics to assess performance and compare it against the existing selected literature's evaluation metrics. The models demonstrated noteworthy efficacy in detecting and recognising masked faces. Despite comparisons with existing literature, the proposed models emerged as the leading contender, providing the best scores across the respective evaluation metrics. This achievement establishes the models' proficiency and effectiveness in the domain of FMD and MFR.

6.2. Conclusion

The proposed study introduces hybrid CNN models that combine various techniques to achieve superior results during model performance evaluations. Furthermore, the proposed models include real-time testing for FMD and MFR to examine and highlight their robustness and ability to generalise in real-time scenarios. The models have demonstrated high accuracies, proving their effectiveness in real-time applications. While only a few existing literature models have tested these scenarios in real-time, the proposed models' successful implementation underscores their significant contribution to the field. Therefore, the proposed advanced and robust CNN models have significantly improved accuracy in both simulated and real-time settings, enhancing performance in challenging scenarios. These scenarios include dimmed-light conditions, extreme dark lighting conditions, fake mask scenarios, incorrect use of a face mask, closed eye scenarios, occlusions of the face, object obstructions of the facial region, angled facial presentations, and computational complexity.

Hence, these models present a versatile solution that can be effectively employed by a wide range of organisations, including companies, schools, universities, hospitals, and the banking sectors that require the implementation and utilisation of an FMD and MFR model. It therefore serves as a valuable tool for implementing robust FMD and MFR functionalities into existing applications or enhancing current biometric systems. The models' adaptability and accuracy make it an ideal choice for diverse sectors seeking to ensure compliance with future and current mask-wearing protocols and to enhance security measures with advanced masked facial recognition capabilities.

6.3. Future Work

Upon the successful development of the FMD and MFR models, it is crucial to recognise and acknowledge the limitations identified within the proposed study. The first limitation identified took place during real-time recognition using the developed *'Custom Real Time Masked Face Recognition'* dataset. A misidentification occurred, where the model erroneously produced a single false positive when the lighting conditions were dimmed completely, and the face was beyond a distance of 200 centimetres. This identified limitation may potentially be attributed towards the camera hardware rather than a flaw in the developed hybrid CNN model. For instance, the developed models utilise a built-in laptop webcam, such lenses are typically designed for close-range use. In essence, different cameras such as Closed-Circuit Television (CCTV) systems, may feature lenses with varying focal lengths or zoom capabilities. It is also noted that the FMD model required a total training duration of one hour whilst the MFR model necessitated an overall training duration of one hour and 30 minutes. Despite the aforementioned limitations, the computational approach consistently produced commendable outcomes in the FMD and MFR models upon the culmination of this investigative research study.

The realm of information technology undergoes constant evolution, encompassing advancements in both hardware and software domains. There exists a vast potential for enhancing authentication and access control mechanisms, which hinges on ongoing research, innovation, and the deployment of thoroughly tested products. Subsequent endeavours will revolve around the enhancement of the current FMD and MFR models. Future work aims to improve the training time required for the models. In addition, it will focus on testing the models with different types of cameras, including but not limited to CCTV cameras and thermal cameras to evaluate its performance under varied hardware for image acquisition. Furthermore, the MFR model aims to mitigate, and where possible eliminate the occurrence of misclassifications identified.

REFERENCES

Abbasi, S., Abdi, H. and Ahmadi, A. 2021. A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset. In: Proceedings of *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*. Tehran, Iran, 3-4 March 2021. 1-6.

Agarap, A. F. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* Available: http://arxiv.org/abs/1803.08375 (Accessed 5 December 2022).

Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S. and Yoon, B. 2020. Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN). *Sensors*, 20 (12): 3344.

Akhtar, N. and Ragavendran, U. 2020. Interpretation of intelligence in CNN-pooling processes: a methodological survey. *Neural computing and applications*, 32 (3): 879-898.

Akingbesote, D., Zhan, Y., Maskeliūnas, R. and Damaševičius, R. 2023. Improving Accuracy of Face Recognition in the Era of Mask-Wearing: An Evaluation of a Pareto-Optimized FaceNet Model with Data Preprocessing Techniques. *Algorithms*, 16 (6): 292.

Al-Dmour, H., Tareef, A., Alkalbani, A. M., Hammouri, A. and Alrahmani, B. 2023. Masked Face Detection and Recognition System Based on Deep Learning Algorithms. *Journal of Advances in Information Technology*, 14 (2).

Al-Rammahi, A. H. I. 2022. Face mask recognition system using MobileNetV2 with optimization function. *Applied Artificial Intelligence*, 36 (1): 2145638.

Albawi, S., Mohammed, T. A. and Al-Zawi, S. 2017. Understanding of a convolutional neural network. In: Proceedings of *2017 International Conference on Engineering and Technology (ICET)*. Antalya, Turkey, 21-23 Aug. 2017. 1-6.

AlGerafi, M. A., Zhou, Y., Oubibi, M. and Wijaya, T. T. 2023. Unlocking the potential: A comprehensive evaluation of augmented reality and virtual reality in education. *Electronics*, 12 (18): 3953.

Alzu'bi, A., Albalas, F., AL-Hadhrami, T., Younis, L. B. and Bashayreh, A. 2021. Masked Face Recognition Using Deep Learning: A Review. *Electronics*, 10 (21): 2666.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M. and Farhan, L. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big data*, 8: 1-74.

Andrade, C. 2020. Understanding the basics of meta-analysis and how to read a forest plot: as simple as it gets. *The Journal of clinical psychiatry*, 81 (5): 21858.

Andrews, I. and Kasy, M. 2019. Identification of and correction for publication bias. *American Economic Review*, 109 (8): 2766-2794.

Arora, C., Gupta, R. and Sridhar, D. S. 2022. Paper presented at the *Face Mask Detection using Deep Learning CNN Architecture*. Singapore, November 2022. 669–677. Available: https://link.springer.com/chapter/10.1007/978-981-19-8825-7_57 (Accessed 01 Janurary 2023).

Arya, Z. A. and Tiwari, V. 2021. An implementation of real-time automatic masked and unmasked face recognition using lbph algorithm. *ADBU Journal of Engineering Technology*, 10 (1).

Aryotejo, G., Adi, P. W., Ernawan, F. and Mufadhol, M. 2023. Detecting and counting coin using opencv and watershed algorithm. In: Proceedings of *AIP Conference Proceedings*. Semarang, Indonesia, 14–15 October 2021.

Ashin, A., Rathika, P. D., Mahavidhya, Y., Hemapriya, N. and Gowri, S. S. 2021. Image Classification in the Era of Deep Learning. In: Proceedings of *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*. Coimbatore, India, 8-9 Oct. 2021. 1-5.

Aswal, V., Tupe, O., Shaikh, S. and Charniya, N. N. 2020. Single Camera Masked Face Identification. In: Proceedings of *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Miami, FL, USA, 14-17 Dec. 2020. 57-60.

Ayyadevara, V. K. 2018. Convolutional Neural Network. In: *Pro Machine Learning Algorithms : A Hands-On Approach to Implementing Algorithms in Python and R*. Berkeley, CA: Apress, 179-215. Available: https://link.springer.com/book/10.1007/978-1-4842-3564-5 (Accessed 02 July 2022).

Bagui, S., Nandi, D., Bagui, S. and White, R. J. 2021. Machine learning and deep learning for phishing email classification using one-hot encoding. *Journal of Computer Science*, 17 (7): 610-623.

Balasubaramanian, S., Cyriac, R., Roshan, S., Paramasivam, K. M. and Jose, B. C. 2023. An effective stacked autoencoder based depth separable convolutional neural network model for face mask detection. *Array*, 19 (100-294).

Balasubramanian, M., Ramyadevi, K. and Geetha, R. 2023. Deep transfer learning based real time face mask detection system with computer vision. *Multimedia Tools and Applications*, 83: 17511–17530.

Bania, R. K. 2023. Ensemble of deep transfer learning models for real-time automatic detection of face mask. *Multimedia Tools and Applications*, 82 (16): 25131-25153.

Bejani, M. M. and Ghatee, M. 2021. A systematic review on overfitting control in shallow and deep neural networks. *Artificial intelligence review*, 54 (8): 6391-6438.

Benifa, J. B., Chola, C., Muaad, A. Y., Hayat, M. A. B., Bin Heyat, M. B., Mehrotra, R., Akhtar, F., Hussein, H. S., Vargas, D. L. R. and Castilla, Á. K. 2023. FMDNet: An Efficient System for Face Mask Detection Based on Lightweight Model during COVID-19 Pandemic in Public Areas. *Sensors*, 23 (13): 6090.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K. and Ghayvat, H. 2021. CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, 10 (20): 2470.

Borenstein, M. 2022. Comprehensive meta-analysis software. In: *Systematic reviews in health research: meta-analysis in context*. 535-548. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119099369.ch27 (Accessed 20 August 2023).

Borenstein, M., Cooper, H., Hedges, L. and Valentine, J. 2019. Heterogeneity in meta-analysis. *The handbook of research synthesis and meta-analysis*, 3: 453-470.

Boulos, M. M. 2021. Facial recognition and face mask detection using machine learning techniques. Master of Science (MS), Montclair State University. Available: https://digitalcommons.montclair.edu/etd/728 (Accessed 05 July 2022).

Brownlee, J. 2020. *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python* Machine Learning Mastery. Available: https://www.google.co.za/books/edition/Data_Preparation_for_Machine_Learning/uAPuDwAAQBAJ?hl=en&gbpv=1&dq=data+preparation (Accessed 19 July 2023).

Cabunilas, J. J. L., Ponio, K. J. P., Villaverde, J. F., Dofitas, B. L. and Juanatas, R. A. 2023. Detection of Acid-Fast Bacilli AFB in Slit-Skin Smear Microscopy using MASK R-CNN Algorithm. In: Proceedings of *2023 13th International Conference on Biomedical Engineering and Technology*. Tokyo, Japan, 30-36.

Cahyono, F., Wirawan, W. and Rachmadi, R. F. 2020. Face Recognition System using Facenet Algorithm for Employee Presence. In: Proceedings of *2020 4th International Conference on Vocational Education and Training (ICOVET)*. Malang, Indonesia, 19-19 Sept. 2020. 57-62.

Cao, D., Dai, R., Wang, J., Ji, B., Alfarraj, O., Tolba, A., Sharma, P. K. and Zhu, M. 2023. Fast visual tracking with squeeze and excitation region proposal network. *Human-centric Computing and Information Sciences*, 13: 1-20.

Cao, P., Zhou, F., Song, Q. and Yang, L. 2024. *Controllable Generation with Text-to-Image Diffusion Models: A Survey*. Available: https://arxiv.org/abs/2403.04279 (Accessed 01 April 2024).

Cascianelli, S., Bello-Cerezo, R., Bianconi, F., Fravolini, M. L., Belal, M., Palumbo, B. and Kather, J. N. 2018. Dimensionality Reduction Strategies for CNN-Based Classification of Histopathological Images. In: De Pietro, G., Gallo, L., Howlett, R. J. and Jain, L. C. eds. Proceedings of *Intelligent Interactive Multimedia Systems and Services 2017*. Cham, Germany 2018. Springer International Publishing, 21-30.

Chakma, B. B., Masud, M., Ahamed, T. and Tusher, M. H. 2022. IDENTIFICATION OF FACE MASK USING CONVOLUTIONAL NEURAL NETWORK-BASED EFFICIENTNET MODEL. *Khulna University Studies*, 1 (Special Issue): 531-538.

Chavda, A., Dsouza, J., Badgujar, S. and Damani, A. 2020. *Multi-Stage CNN Architecture for Face Mask Detection*.

Chen, J., Zhang, X., Tang, Y. and Yu, H. 2023. ICE-YoloX: research on face mask detection algorithm based on improved YoloX network. *The Journal of Supercomputing*, Volume 80, : 2948–2969.

Chen, T., Moreau, T., Jiang, Z., Zheng, L., Yan, E., Cowan, M., Shen, H., Wang, L., Hu, Y. and Ceze, L. 2018. TVM: An automated end-to-end optimizing compiler for deep learning. In: Proceedings of *13th USENIX Symposium on Operating Systems Design and Implementation 5*. Carlsbad, CA, USA, 8–10 October 2018. 578-594.

Chen, Y., Cui, L. and Ding, M. 2023. Neural Processing of Affective Scenes: A Comparison between Convolutional Neural Networks and Human Visual Pathways. *Journal of Vision*, 23 (9): 5044-5044.

Chin, T.-W., Chuang, P. I.-J., Chandra, V. and Marculescu, D. 2020. One weight bitwidth to rule them all. In: Proceedings of *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 85-103.

Christa, G. H., Jesica, J., Anisha, K. and Sagayam, K. M. 2021. CNN-based mask detection system using openCV and MobileNetV2. In: Proceedings of *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*. Coimbatore, India, 13-14 May 2021. IEEE, 115-119.

Chunming, W. and Ying, Z. 2021. MTCNN and FACENET Based Access Control System for Face Detection and Recognition. *Automatic Control and Computer Sciences*, 55 (1): 102-112.

Cimmino, L., Nappi, M., Narducci, F. and Pero, C. 2022. M2FRED: Mobile masked face REcognition through periocular dynamics analysis. *IEEE Access*, 10: 94388-94402.

Damer, N., Boutros, F., Süßmilch, M., Fang, M., Kirchbuchner, F. and Kuijper, A. 2021. *Masked face recognition: Human vs. machine*. Available: https://arxiv.org/abs/2103.01924 (Accessed 03 October 2022).

Dan, B., Sun, X. and Liu, L. 2019. Diseases and pests identification of Lycium barbarum using se-mobilenet v2 algorithm. In: Proceedings of *2019 12th International symposium on computational intelligence and design (ISCID)*. Hangzhou, China, 14-15 December 2019. IEEE, 121-125.

Darborg, A. 2020. Real-time face recognition using one-shot learning: A deep learning and machine learning project. Master of Science in Computer Engineering, Mid Sweden University Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1474723&dswid=4215 (Accessed

Das, A., Ansari, M. W. and Basak, R. 2020. Covid-19 face mask detection using TensorFlow, Keras and OpenCV. In: Proceedings of *2020 IEEE 17th India Council International Conference (INDICON)*. New Delhi, India, 10-13 December 2020. IEEE, 1-5.

Deng, H., Feng, Z., Qian, G., Lv, X., Li, H. and Li, G. 2021. MFCosface: a masked-face recognition algorithm based on large margin cosine loss. *Applied Sciences*, 11 (16): 7310.

Deng, X., Liu, Q., Deng, Y. and Mahadevan, S. 2016. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, 340: 250-261.

Desai, A. N. and Mehrotra, P. 2020. Medical Masks. *JAMA*, 323 (15): 1517-1518.

Dharanesh, S. and Rattani, A. 2021. Post-COVID-19 mask-aware face recognition system. In: Proceedings of *2021 IEEE International Symposium on Technologies for Homeland Security (HST)*. Boston, MA, USA, 08-09 November 2021. IEEE, 1-7.

Ding, Y., Zhou, Y., Zhu, Y., Ye, Q. and Jiao, J. 2019. Selective sparse sampling for fine-grained image recognition. In: Proceedings of *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Shenzhen, China, 27 October-2 November 2019. 6599-6608.

Dong, K., Zhou, C., Ruan, Y. and Li, Y. 2020. MobileNetV2 model for image classification. In: Proceedings of *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. Guangzhou, China, 18-20 December 2020. IEEE, 476-480.

Dürr, O., Sick, B. and Murina, E. 2020. *Probabilistic deep learning: With python, keras and tensorflow probability* Manning Publications. Available: https://www.google.co.za/books/edition/Probabilistic_Deep_Learning/-bYCEAAAQBAJ?hl=en&gbpv=1&dq=probabilistic+deep+learning (Accessed 15 July 2023).

Ertuğrul, Ö. F. 2018. A novel type of activation function in artificial neural networks: Trained activation function. *Neural Networks*, 99: 148-157.

Fazeli Ardekani, P., Tale, S. Z. and Parseh, M. J. 2023. Face mask recognition using a custom CNN and data augmentation. *Signal, Image and Video Processing*, 18: 255–263.

Francone, M., Budde, R. P., Bremerich, J., Dacher, J. N., Loewe, C., Wolf, F., Natale, L., Pontone, G., Redheuil, A. and Vliegenthart, R. 2020. CT and MR imaging prior to transcatheter aortic valve implantation: standardisation of scanning protocols, measurements and reporting—a consensus document by the European Society of Cardiovascular Radiology (ESCR). *European radiology*, 30: 2627-2650.

Gholamalinezhad, H. and Khosravi, H. 2020. Pooling methods in deep neural networks, a review. *arXiv preprint arXiv:2009.07485*, Article ID.

Głowacka, N. and Rumiński, J. 2021. Face with mask detection in thermal images using deep neural networks. *Sensors*, 21 (19): 1-16.

Godavitarne, C., Robertson, A., Ricketts, D. M. and Rogers, B. A. 2018. Understanding and interpreting funnel plots for the clinician. *British Journal of Hospital Medicine*, 79 (10): 578-583.

Grandini, M., Bagli, E. and Visani, G. 2020. *Metrics for multi-class classification: an overview*. Available: https://arxiv.org/abs/2008.05756 (Accessed 21 October 2023).

Granger, B. E. and Pérez, F. 2021. Jupyter: Thinking and storytelling with code and data. *Computing in Science & Engineering*, 23 (2): 7-14.

Guo, S., Li, L., Guo, T., Cao, Y. and Li, Y. 2022. Research on mask-wearing detection algorithm based on improved YOLOv5. *Sensors*, 22 (13): 1-16.

Gurav, O. 2020. *Face Mask Detection Dataset*. Available: https://www.kaggle.com/datasets/omkargurav/face-mask-dataset (Accessed 30 July 2023).

Habib, S., Alsanea, M., Aloraini, M., Al-Rawashdeh, H. S., Islam, M. and Khan, S. 2022. An Efficient and Effective Deep Learning-Based Model for Real-Time Face Mask Detection. *Sensors*, 22 (7): 1-13.

Hao, W., Yizhou, W., Yaqin, L. and Zhili, S. 2020. The role of activation function in CNN. In: Proceedings of *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. Guangzhou, China, 18-20 December 2020. IEEE, 429-432.

Hariri, W. 2022. Efficient masked face recognition method during the covid-19 pandemic. *Signal, Image and Video Processing*, 16 (3): 605-612.

Hodges, C. B., Stone, B. M., Johnson, P. K., Carter III, J. H., Sawyers, C. K., Roby, P. R. and Lindsey, H. M. 2023. Researcher degrees of freedom in statistical software contribute to unreliable results: a comparison of nonparametric analyses conducted in SPSS, SAS, Stata, and R. *Behavior Research Methods*, 55 (6): 2813-2837.

Howse, J. and Minichino, J. 2020. *Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning* Packt Publishing Ltd. Available: https://www.google.co.za/books/edition/Learning_OpenCV_4_Computer_Vision_with_P/ef_RDwAAQBAJ?hl=en&gbpv=1&dq=learning+opencv+4+computer+vision (Accessed 06 September 2022).

Hu, Y., Xu, Y., Zhuang, H., Weng, Z. and Lin, Z. 2022. Machine Learning Techniques and Systems for Mask-Face Detection—Survey and a New OOD-Mask Approach. *Applied Sciences*, 12 (18): 1-37.

Hussain, D., Ismail, M., Hussain, I., Alroobaea, R., Hussain, S. and Ullah, S. S. 2022. Face mask detection using deep convolutional neural network and MobileNetV2-based transfer learning. *Wireless Communications and Mobile Computing*, 2022: 1-10.

Hussain, S., Yu, Y., Ayoub, M., Khan, A., Rehman, R., Wahid, J. A. and Hou, W. 2021. IoT and deep learning based approach for rapid screening and face mask detection for infection spread control of COVID-19. *Applied Sciences*, 11 (8): 1-27.

Ilyas, Q. M. and Ahmad, M. 2022. An Enhanced Deep Learning Model for Automatic Face Mask Detection. *Intelligent Automation & Soft Computing*, 31 (1).

Islam, M. S., Moon, E. H., Shaikat, M. A. and Alam, M. J. 2020. A novel approach to detect face mask using CNN. In: Proceedings of *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*. Thoothukudi, India, 03-05 December 2020. IEEE, 800-806.

Jangra, A. 2020. *Face Mask Detection ~12K Images Dataset*. Available: https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset (Accessed 30 July 2023).

Jayaswal, R. and Dixit, M. 2020. Comparative Analysis of Human Face Recognition by Traditional Methods and Deep Learning in Real-Time Environment. In: Proceedings of *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*. Gwalior, India, 10-12 April 2020. 66-71.

Jayaswal, R. and Dixit, M. 2023. AI-based face mask detection system: a straightforward proposition to fight with Covid-19 situation. *Multimedia Tools and Applications*, 82 (9): 13241-13273.

Jiang, E. 2020. A review of the comparative studies on traditional and intelligent face recognition methods. In: Proceedings of *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*. Chongqing, China, 10-12 July 2020. 11-15.

Jogin, M., Mohana, Madhulika, M. S., Divya, G. D., Meghana, R. K. and Apoorva, S. 2018. Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning. In: Proceedings of *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. Bangalore, India, 18-19 May 2018. 2319-2323.

Joseph, V. R. and Vakayil, A. 2022. SPlit: An Optimal Method for Data Splitting. *Technometrics*, 64 (2): 166-176.

Kakarla, S., Gangula, P., Rahul, M. S., Singh, C. S. C. and Sarma, T. H. 2020. Smart Attendance Management System Based on Face Recognition Using CNN. In: Proceedings of *2020 IEEE-HYDCON*. Hyderabad, India, 11-12 Sept. 2020. 1-5.

Kaloev, M. and Krastev, G. 2021. Comparative Analysis of Activation Functions Used in the Hidden Layers of Deep Neural Networks. In: Proceedings of *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. Ankara, Turkey, 11-13 June 2021. 1-5.

Kamil, M. H. M., Zaini, N., Mazalan, L. and Ahamad, A. H. 2023. Online attendance system based on facial recognition with face mask detection. *Multimedia Tools and Applications*, 82: 34437–34457.

Kaur, G., Sinha, R., Tiwari, P. K., Yadav, S. K., Pandey, P., Raj, R., Vashisth, A. and Rakhra, M. 2022. Face mask recognition system using CNN model. *Neuroscience Informatics*, 2 (3): 1-9.

Kavitha, M. N., Kanimozhi, N., Saranya, S. S., Sri, S. J., Kalpana, V. and Jayavarthiniy, K. 2022. Face Mask Detection Using Deep Learning. In: Proceedings of *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. Coimbatore, India, 23-25 Feb. 2022. 319-324.

Kaziakhmedov, E., Kireev, K., Melnikov, G., Pautov, M. and Petiushko, A. 2019. Real-world Attack on MTCNN Face Detection System. In: Proceedings of *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*. Novosibirsk, Russia, 21-27 Oct. 2019. 0422-0427.

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A. S. 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53: 5455-5516.

Khandelwal, S., Prasad, A., Kumar, A., Gautam, J. and Patle, A. 2023. A Study on Efficient Image Classification of Historical Monuments Using CNN. In: Proceedings of *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*. Coimbatore, India, 03-05 August 2023. IEEE, 220-225.

Kheaksong, A., Samothai, P., Sanguansat, P., Ngamloed, T., Srisomboon, K. and Lee, W. 2022. Evaluation of Masked Face Recognition of FaceNet Implemented with Machine Learning Algorithms. In: Proceedings of *2022 6th International Conference on Information Technology (InCIT)*. Nonthaburi, Thailand, 10-11 Nov. 2022. 475-478.

Kirana, K. C., Wibawanto, S., Hidayah, N., Cahyono, G. P. and Asfani, K. 2019. Improved Neural Network using Integral-RELU based Prevention Activation for Face Detection. In: Proceedings of *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*. Denpasar, Indonesia, 3-4 Oct. 2019. 260-263.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D. J. 2021. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151: 1-21.

Kodros, J. K., O'Dell, K., Samet, J. M., L'Orange, C., Pierce, J. R. and Volckens, J. 2021. Quantifying the health benefits of face masks and respirators to mitigate exposure to severe air pollution. *GeoHealth*, 5 (9): 1-14.

Kong, C. and Lucey, S. 2017. *Take it in your stride: Do we need striding in CNNs?* Available: https://arxiv.org/abs/1712.02502 (Accessed 23 March 2023).

Kortli, Y., Jridi, M., Al Falou, A. and Atri, M. 2020. Face recognition systems: A survey. *Sensors*, 20 (2): 342.

Kumar, A. 2023. A cascaded deep-learning-based model for face mask detection. *Data Technologies and Applications*, 57 (1): 84-107.

Kumar, A., Kalia, A. and Kalia, A. 2022. ETL-YOLO v4: A face mask detection algorithm in era of COVID-19 pandemic. *Optik*, 259: 169051.

Kumar, A., Kalia, A., Verma, K., Sharma, A. and Kaushal, M. 2021. Scaling up face masks detection with YOLO on a novel dataset. *Optik*, 239: 1-15.

Kumar, B. A. and Bansal, M. 2023. Face mask detection on photo and real-time video images using Caffe-MobileNetV2 transfer learning. *Applied Sciences*, 13 (2): 1-22.

Kumar, S., Yadav, D., Gupta, H., Kumar, M. and Verma, O. P. 2023. Towards smart surveillance as an aftereffect of COVID-19 outbreak for recognition of face masked individuals using YOLOv3 algorithm. *Multimedia Tools and Applications*, 82 (6): 8381-8403.

Kwak, N. and Kim, D. 2023. Detection of Worker's Safety Helmet and Mask and Identification of Worker Using Deeplearning. *CMC-COMPUTERS MATERIALS & CONTINUA*, 75 (1): 1671-1686.

Li, H.-C., Deng, Z.-Y. and Chiang, H.-H. 2020. Lightweight and resource-constrained learning network for face recognition with performance optimization. *Sensors*, 20 (21): 1-20.

Lian, Z., Wang, H. and Zhang, Q. 2022. An Image Deblurring Method Using Improved U-Net Model. *Mobile Information Systems*, 2022: 1-11.

Liang, M., Gao, L., Cheng, C., Zhou, Q., Uy, J. P., Heiner, K. and Sun, C. 2020. Efficacy of face mask in preventing respiratory virus transmission: A systematic review and meta-analysis. *Travel Medicine and Infectious Disease*, 36: 1-9.

Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P. C., Ioannidis, J. P., Clarke, M., Devereaux, P. J., Kleijnen, J. and Moher, D. 2009. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *Annals of internal medicine*, 151 (4): 65-94.

Lipworth, B., Chan, R. and Kuo, C. R. 2020. COVID-19: Start with the nose. *Journal of Allergy and Clinical Immunology*, 146 (5): 1-2.

Loey, M., Manogaran, G., Taha, M. H. N. and Khalifa, N. E. M. 2021. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement*, 167: 1-11.

Ma, J. and Yuan, Y. 2019. Dimension reduction of image deep feature using PCA. *Journal of Visual Communication and Image Representation*, 63: 1-8.

Mahmoud, H. A. H., Alharbi, A. H. and Alghamdi, N. S. 2022. A Framework for Mask-Wearing Recognition in Complex Scenes for Different Face Sizes. *Intelligent Automation & Soft Computing*, 32 (2).

Maji, K. and Gupta, S. 2023. Evaluation of Various Loss Functions and Optimization Techniques for MRI Brain Tumor Detection. In: Proceedings of *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. Ballar, India, 29-30 April 2023. IEEE, 1-6.

Mandal, B., Okeukwu, A. and Theis, Y. 2021. *Masked face recognition using resnet-50*. Available: https://arxiv.org/abs/2104.08997 (Accessed 03 February 2023).

Marks-Anglin, A. and Chen, Y. 2020. A historical review of publication bias. *Research synthesis methods*, 11 (6): 725-742.

Marwa, K. and Kais, O. 2022. Classifiers Combination for Efficient Masked Face Recognition. *International Journal of Advanced Computer Science and Applications*, 13 (9).

Masi, I., Trần, A. T., Hassner, T., Sahin, G. and Medioni, G. 2019. Face-Specific Data Augmentation for Unconstrained Face Recognition. *International Journal of Computer Vision*, 127 (6): 642-667.

Meng, F., Cheng, H., Zhuang, J., Li, K. and Sun, X. 2021. *Rmnet: Equivalently removing residual connection from networks*. Available: https://arxiv.org/abs/2111.00687 (Accessed 5 May 2023).

Mhadgut, S. 2021. Masked Face Detection and Recognition System in Real Time using YOLOv3 to combat COVID-19. In: Proceedings of *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Kharagpur, India, 29-30 April 2023. IEEE, 1-7.

Mohammed Ali, F. A. and Al-Tamimi, M. S. 2022. Face mask detection methods and techniques: A review. *International Journal of Nonlinear Analysis and Applications*, 13 (1): 3811-3823.

Mohan, B. P. and Adler, D. G. 2019. Heterogeneity in systematic review and meta-analysis: how to read between the numbers. *Gastrointestinal Endoscopy*, 89 (4): 902-903.

Moradi, R., Berangi, R. and Minaei, B. 2020. A survey of regularization strategies for deep models. *Artificial intelligence review*, 53 (6): 3947-3986.

Muhammad, A. R., Utomo, H. P., Hidayatullah, P. and Syakrani, N. 2022. Early Stopping Effectiveness for YOLOv4. *Journal of Information Systems Engineering & Business Intelligence*, 8 (1).

Mundial, I. Q., Hassan, M. S. U., Tiwana, M. I., Qureshi, W. S. and Alanazi, E. 2020. Towards Facial Recognition Problem in COVID-19 Pandemic. In: Proceedings of *2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*. Medan, Indonesia, 3-4 Sept. 2020. 210-214.

Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P. and Hemanth, J. 2021. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable cities and society*, 66: 102692.

Naseri, H. and Mehrdad, V. 2023. Novel CNN with investigation on accuracy by modifying stride, padding, kernel size and filter numbers. *Multimedia Tools and Applications*, Article ID.

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., Malík, P. and Hluchý, L. 2019. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial intelligence review*, 52 (1): 77-124.

Nuruddin Qaisar Bhuiyan, M., Shamsujjoha, M., Ripon, S. H., Proma, F. H. and Khan, F. 2019. Chapter 4 - Transfer Learning and Supervised Classifier Based Prediction Model for Breast Cancer. In: Dey, N., Das, H., Naik, B. and Behera, H. S. eds. *Big Data Analytics for Intelligent Healthcare Management*. Academic Press, 59-86. Available: https://www.sciencedirect.com/science/article/pii/B9780128181461000040 (Accessed 10 October 2021).

Oumina, A., Makhfi, N. E. and Hamdi, M. 2020. Control The COVID-19 Pandemic: Face Mask Detection Using Transfer Learning. In: Proceedings of *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. Kenitra, Morocco, 2-3 Dec. 2020. 1-5.

Page, M. J., Sterne, J. A., Higgins, J. P. and Egger, M. 2021. Investigating and dealing with publication bias and other reporting biases in meta-analyses of health research: A review. *Research synthesis methods*, 12 (2): 248-259.

Pang, B., Nijkamp, E. and Wu, Y. N. 2020. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45 (2): 227-248.

Pann, V. and Lee, H. J. 2022. Effective attention-based mechanism for masked face recognition. *Applied Sciences*, 12 (11): 1-14.

Peng, H., Xing, Z., Liu, X., Gao, Z. and He, H. 2023. Toward masked face recognition: An effective facial feature extraction and refinement model in multiple scenes. *Expert Systems*, 40 (2): 1-18.

Qi, C. and Yang, L. 2020. Face recognition in the scene of wearing a mask. In: Proceedings of *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*. Ottawa, ON, Canada, 12-13 Sept. 2020. 77-80.

Qinjun, L., Tianwei, C., Yan, Z. and Yuying, W. Facial Recognition Technology: A Comprehensive Overview. *Academic Journal of Computing & Information Science*, 6 (7): 15-26.

Rafidison, M. A., Rakotomihamina, A. H., Rafanantenana, S. H. J., Toky, R. F. M., Raoelina, M. M. N. and Ramafiarisona, H. M. 2023. Neural networks contribution in face mask detection to reduce the spread of COVID-19. *Multimedia Tools and Applications*, 82: 32559–32581.

Rahmani, M. K. I., Taranum, F., Nikhat, R., Farooqi, M. R. and Khan, M. A. 2022. Automatic Real-Time Medical Mask Detection Using Deep Learning to Fight COVID-19. *Comput. Syst. Sci. Eng.*, 42 (3): 1181-1198.

Reddy, N. V. M. C. and Kumar, K. 2021. Comparison of HOG and Fisherfaces Based Face Recognition System Using MATLAB. In: Proceedings of *2021 2nd International Conference for Emerging Technology (INCET)*. Belagavi, India, 21-23 May 2021. 1-5.

Rokhana, R., Herulambang, W. and Indraswari, R. 2021. Multi-class image classification based on mobilenetv2 for detecting the proper use of face mask. In: Proceedings of *2021 International Electronics Symposium (IES)*. Surabaya, Indonesia, 29-30 September 2021. IEEE, 636-641.

Saleem, S., Shiney, J., Shan, B. P. and Mishra, V. K. 2023. Face recognition using facial features. In: Proceedings of *2023 International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology(5NANO)*. Punjab, India 3857-3862.

Salehi, W., Baglat, P., Gupta, G., Khan, S. B., Almusharraf, A., Alqahtani, A. and Kumar, A. 2023. An approach to binary classification of alzheimer's disease using lstm. *Bioengineering*, 10 (8): 1-22.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of *2018 Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*. Salt Lake City, UT, USA, 18-23 June 2018. 4510-4520.

Sarkar, A. and Singh, B. K. 2020. A review on performance, security and various biometric template protection schemes for biometric authentication systems. *Multimedia Tools and Applications*, 79: 27721-27776.

Sarkis-Onofre, R., Catalá-López, F., Aromataris, E. and Lockwood, C. 2021. How to properly use the PRISMA Statement. *Systematic Reviews*, 10 (1): 1-3.

Seng, S., Al-Ameen, M. N. and Wright, M. 2021. A first look into users' perceptions of facial recognition in the physical world. *Computers & Security*, 105: 1-24.

Septi, S. Q. N., Yulita, I. N. and Napitupulu, H. 2021. Face Recognition Using Fisherface and Support Vector Machine Method. In: Proceedings of *2021 International Conference on Artificial Intelligence and Big Data Analytics*. Bandung, Indonesia, 27-29 October 2021. IEEE, 50-55.

Serengil, S. I. and Ozpinar, A. 2020. Lightface: A hybrid deep face recognition framework. In: Proceedings of *2020 innovations in intelligent systems and applications conference (ASYU)*. Istanbul, Turkey, 15-17 October 2020. IEEE, 1-5.

Sergeev, A. and Del Balso, M. 2018. *Horovod: fast and easy distributed deep learning in TensorFlow*. Available: https://arxiv.org/abs/1802.05799 (Accessed 10 February 2022).

Shams, M. S. 2023. *Role of Activation functions in Neural Networks*. Available: https://www.linkedin.com/pulse/role-activation-functions-neural-networks-md-salman-shams (Accessed 15 March 2023).

Sharma, V. 2020. Face mask detection using yolov5 for COVID-19. Master of Computer Science, California State University San Marcos. Available: https://scholarworks.calstate.edu/downloads/wp988p69r (Accessed 10 December 2022).

Sheikh, B. U. h. and Zafar, A. 2023. Untargeted white-box adversarial attack to break into deep learning based COVID-19 monitoring face mask detection system. *Multimedia Tools and Applications*, 83: 23873–23899.

Sheikh, S., Shendre, S., Anjum, F., Qureshi, M., Waghchoure, S. and Assudani, M. 2018. Face recognition using cnn. *International Journal for Research in Applied Science and Engineering Technology*, 6 (3): 1411-1414.

Shorten, C. and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6 (1): 1-48.

Singh, A. 2023a. *Masked Face Recognition Dataset*. Available: https://www.kaggle.com/datasets/nanimasaka/masked-face-recognition-dataset (Accessed 30 July 2023).

Singh, S. K. 2023b. *CNN Basics and Evolution*. Available: https://www.linkedin.com/pulse/cnn-basics-evolution-sateesh-kumar-singh (Accessed 12 September 2023).

Sitepu, S. E., Jati, G., Alhamidi, M. R., Caesarendra, W. and Jatmiko, W. 2021. FaceNet with RetinaFace to Identify Masked Face. In: Proceedings of *2021 6th International Workshop on Big Data and Information Security (IWBIS)*. Depok, Indonesia, 23-25 October 2021. IEEE, 81-86.

Staff, K. 2021. *An engineer's guide to deploying machine learning in smart devices using audio edge processors*. Available: https://www.wevolver.com/article/an-engineers-guide-to-deploying-machine-learning-in-smart-devices-using-audio-edge-processors (Accessed 31 July 2023).

StataCorp. 2023. Stata Statistical Software: Release 18. (computer software).

Suhaimin, M. S. M., Hijazi, M. H. A., Kheau, C. S. and On, C. K. 2021. Real-time mask detection and face recognition using eigenfaces and local binary pattern histogram for attendance system. *Bulletin of Electrical Engineering and Informatics*, 10 (2): 1105-1113.

Sun, M., Song, Z., Jiang, X., Pan, J. and Pang, Y. 2017. Learning Pooling for Convolutional Neural Network. *Neurocomputing*, 224: 96-104.

Szeliski, R. 2022. *Computer vision: algorithms and applications* Springer Nature. Available: https://www.google.co.za/books/edition/Computer_Vision/QptXEAAAQBAJ?hl=en&gbpv=1&dq=computer+vision+algorithms+and+applications (Accessed 10 Janurary 2023).

Talahua, J. S., Buele, J., Calvopiña, P. and Varela-Aldás, J. 2021. Facial recognition system for people with and without face mask in times of the covid-19 pandemic. *Sustainability*, 13 (12): 6900.

Teboulbi, S., Messaoud, S., Hajjaji, M. A. and Mtibaa, A. 2021. Real-time implementation of AI-based face mask detection and social distancing measuring system for COVID-19 prevention. *Scientific Programming*, 2021: 1-21.

Toppo, K., Kumar, N., Kumar, P. and Tanwar, L. 2022. Face Mask Detection Using MobileNetV2 and Implementation Using Different Face Detectors. In: Proceedings of *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021*. Singapore., Springer, 287-303.

Ullah, N., Javed, A., Ghazanfar, M. A., Alsufyani, A. and Bourouis, S. 2022. A novel DeepMaskNet model for face mask detection and masked facial recognition. *Journal of King Saud University-Computer and Information Sciences*, 34 (10): 9905-9914.

Umer, M., Sadiq, S., Alhebshi, R. M., Alsubai, S., Al Hejaili, A., Nappi, M. and Ashraf, I. 2023. Face mask detection using deep convolutional neural network and multi-stage image processing. *Image and Vision Computing*, 133: 1-10.

Üzen, H., Turkoglu, M., Aslan, M. and Hanbay, D. 2023. Depth-wise Squeeze and Excitation Block-based Efficient-Unet model for surface defect detection. *The Visual Computer*, 39 (5): 1745-1764.

Vadlapati, J., Velan, S. S. and Varghese, E. 2021. Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic. In: Proceedings of *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Kharagpur, India, 6-8 July 2021. 1-5.

Van Aert, R. C., Wicherts, J. M. and Van Assen, M. A. 2019. Publication bias examined in meta-analyses from psychology and medicine: A meta-meta-analysis. *PloS one*, 14 (4): 1-32.

van den Berg, Y. H., Lansu, T. A. and Cillessen, A. H. 2020. Preference and popularity as distinct forms of status: A meta-analytic review of 20 years of research. *Journal of Adolescence*, 84: 78-95.

Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R. and Mittal, A. 2019. Pneumonia Detection Using CNN based Feature Extraction. In: Proceedings of *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Coimbatore, India, 20-22 Feb. 2019. 1-7.

Wang, K., Chen, C. and He, Y. 2020. Research on pig face recognition model based on keras convolutional neural network. In: Proceedings of *IOP Conference Series: Earth and Environmental Science*. Tianjin, China, April 2020. (3) : 1-11.

Wang, M., Sun, K. and Guo, A. 2023. Wheat ear detection using anchor-free ObjectBox model with attention mechanism. *Signal, Image and Video Processing*, 17 (7): 3425-3432.

Wang, Y., Li, Y. and Zou, H. 2023. Masked Face Recognition System Based on Attention Mechanism. *Information*, 14 (2): 87.

Wang, Y., Sun, D., Chen, K., Lai, F. and Chowdhury, M. 2023. Egeria: Efficient dnn training with knowledge-guided layer freezing. In: Proceedings of *Proceedings of the Eighteenth European Conference on Computer Systems*. Sai Kung, Hong Kong 8-12 May 2023. 851-866.

Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N. and Pei, Y. 2020. *Masked face recognition dataset and application*. Available: https://arxiv.org/abs/2003.09093 (Accessed 31 July 2022).

Weerts, H. J., Mueller, A. C. and Vanschoren, J. 2020. Importance of tuning hyperparameters of machine learning algorithms. *arXiv preprint arXiv:2007.07588*, Article ID.

Wei, X., Feng, W., Lei, Q., Gui, G., Le, S., Zhong, J. and Wang, W. 2020. Defect Detection of Using Variant CNN in the Processing of Cover Glass, Touch Screen and Display under Parallel Light. In: Proceedings of *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. Chengdu, China, 11-14 Dec. 2020. 1349-1355.

WHO, W. H. O. 2023. *WHO Coronavirus (COVID-19)*. Available: https://covid19.who.int/ (Accessed 14 March 2023).

William, I., Setiadi, D. R. I. M., Rachmawanto, E. H., Santoso, H. A. and Sari, C. A. 2019. Face Recognition using FaceNet (Survey, Performance Test, and Comparison). In: Proceedings of *2019 Fourth International Conference on Informatics and Computing (ICIC)*. Semarang, Indonesia, 16-17 Oct. 2019. 1-6.

Wojcik, M. E. and Austin, D. W. 2020. Criminal justice and Covid-19. *Criminal Justice*, 35 (3): 44-48.

Wu, C. and Zhang, Y. 2021. MTCNN and FACENET based access control system for face detection and recognition. *Automatic Control and Computer Sciences*, 55: 102-112.

Wu, Y. and Yang, L. 2022. Masked Face Recognition Using FaceNet. In: Proceedings of *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*. Nanjing, China, 18-21 Nov. 2022. 716-720.

Xiao, X., Mudiyanselage, T. B., Ji, C., Hu, J. and Pan, Y. 2019. Fast deep learning training through intelligently freezing layers. In: Proceedings of *2019 international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*. Atlanta, GA, USA, 14-17 July 2019. IEEE, 1225-1232.

Xu, X.-F., Zhang, L., Duan, C.-D. and Lu, Y. 2019. Research on inception module incorporated siamese convolutional neural networks to realize face recognition. *IEEE Access*, 8: 12168-12178.

Yang, C.-Y., Samani, H., Ji, N., Li, C., Chen, D.-B. and Qi, M. 2022. Deep Learning Based Real-Time Facial Mask Detection and Crowd Monitoring. 40 (6): 1263-11294.

Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G. and Wang, W. 2020. Face mask recognition system with YOLOV5 based on image recognition. In: Proceedings of *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. Chengdu, China, IEEE, 1398-1404.

Yang, Z., Ge, W. and Zhang, Z. 2020. Face recognition based on MTCNN and integrated application of FaceNet and LBP method. In: Proceedings of *2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*. Manchester, United Kingdom, 15-17 October 2020. IEEE, 95-98.

Yang, Z., Yuan, Y., Zhang, M., Zhao, X., Zhang, Y. and Tian, B. 2019. Safety Distance Identification for Crane Drivers Based on Mask R-CNN. *Sensors*, 19 (12): 2789.

Yepez, J. and Ko, S. B. 2020. Stride 2 1-D, 2-D, and 3-D Winograd for Convolutional Neural Networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28 (4): 853-863.

Yi, D., Ahn, J. and Ji, S. 2020. An effective optimization method for machine learning based on ADAM. *Applied Sciences*, 10 (3): 1073.

Yong, L., Ma, L., Sun, D. and Du, L. 2023. Application of MobileNetV2 to waste classification. *Plos one*, 18 (3): 1-6.

Yu, J. and Zhang, W. 2021. Face mask wearing detection algorithm based on improved YOLO-v4. *Sensors*, 21 (9): 3263.

Yuan, L., Qu, Z., Zhao, Y., Zhang, H. and Nian, Q. 2017. A convolutional neural network based on TensorFlow for face recognition. In: Proceedings of *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 25-26 March 2017. 525-529.

Zebari, R., Mohsin Abdulazeez, A., Zeebaree, D., Zebari, D. and Saeed, J. 2020. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, 1: 56-70.

Zhang, E. 2021. A Real-Time Deep Transfer Learning Model for Facial Mask Detection. In: Proceedings of *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*. Dulles, VA, USA, 19-23 April 2021. 1-7.

Zhang, H., Wang, X., Zhu, J. and Kuo, C. C. J. 2018. Accelerating proposal generation network for fast face detection on mobile devices. In: Proceedings of *2018 25th IEEE International Conference on Image Processing (ICIP)*. Athens, Greece, 07-10 October 2018. IEEE, 326-330.

Zhang, N., Luo, J. and Gao, W. 2020. Research on Face Detection Technology Based on MTCNN. In: Proceedings of *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. Xi'an, China, 25-27 Sept. 2020. 154-158.

Zhang, Y. and Deng, W. 2020. Class-balanced training for deep face recognition. In: Proceedings of *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshps*. Beijing, 824-825.

Zhang, Z., Huang, L., Pauloski, J. G. and Foster, I. T. 2020. Efficient i/o for neural network training with compressed data. In: Proceedings of *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. New Orleans, LA, USA, 18-22 May 2020. IEEE, 409-418.

Zhihuan, W., Xiangning, C., Yongming, G. and Yuntao, L. 2018. Rapid target detection in high resolution remote sensing images using yolo model. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42 (3).

ANNEXURE A: META-ANALYSIS INCLUDED STUDIES

Authors	Publication Year	Study label	Country	Approach	Model Type	Method	Model	No of Data-set	Dataset Names	Sample Size	Acc	Prec	Rec	f1-Score
(Al-Dmour et al. 2023)	2023	Study 1	Jordan	Supervised	fmd	Deep Learning	CNN,VGG-16	3	D1-RMFD D2-MaskedFaceNet D3-CFR	42460	99,50	99,85	99,85	99,85
(Al-Dmour et al. 2023)	2023	Study 2	Jordan	Supervised	mfr	Deep Learning	CNN,VGG-16	3	D1-RMFD D2-MaskedFaceNet D3-CFR	42460	97,98	97,40	98,60	97,60
(Su et al. 2022)	2022	Study 3	China	Supervised	fmd	Transfer Learning and Deep Learning	Efficient-Yolov3 MobileNet	2	D1-MAFA D2-WIDER Face	17811	97,84	98,40	97,87	97,87
(Hu *et al.* 2022)	2022	Study 4	Singapore	Supervised	fmd	Deep Learning	YoloX	2	D1-Wider Face D2-FMD Dataset by Larxel	8114	76,18	83,08	72,65	77,51
(Ilyas and Ahmad 2022)	2022	Study 5	Saudi Arabia	Supervised	fmd	Machine Learning and Deep Learning	MobileNetV2, Logistic Regression	1	Custom dataset(Composed of random public repositories)	3000	100	99,90	99,90	99,90
(Kumar 2023)	2022	Study 6	India	Supervised	fmd	Deep Learning	ResNet34 Yolov SSP	1	Custom dataset	100006	85,16	81,00	65,00	73,00
(Ullah et al. 2022)	2022	Study 7	Pakistan	Supervised	fmd	Deep Learning	DeepMaskNet AlexNet	4	D1-Facemask Dataset by Sushant Shah D2- Facemask by Sumansid D3-Facemask Detection Dataset Lavik Jain D4-Custom Dataset(MDMFR)	1476345	100	100	100	100
(Ullah et al. 2022)	2022	Study 8	Pakistan	Supervised	mfr	Deep Learning	DeepMaskNet AlexNet	4	D1-Facemask Dataset by Sushant Shah D2- Facemask by Sumansid D3-Facemask Detection Dataset Lavik Jain D4-Custom Dataset(MDMFR)	1476345	93,33	93,00	94,50	93,74
(Naseri, Kurnaz and Farhan 2023)	2023	Study 9	Turkey	Supervised	fmd	Deep Learning	CNN ASMFO-HResMobileNet	3	D1-FMD Dataset by Larxel D2-Custom Dataset D3-Custom Dataset (videos collected)	2584	95,25	90,94	95,34	92,99
(Balasubramanian, Ramyadevi and Geetha 2023)	2023	Study 10	India	Supervised	fmd	Deep Learning	ResNet152V2 OpenCV	1	Custom Dataset	14 000 000	99,10	99,20	99,00	99,10
(Hussain et al. 2022)	2022	Study 11	Saudi Arabia	Supervised	fmd	Transfer Learning and Deep Learning	MobileNetV2	2	D1-Custom Dataset D2- PyImage Search Reader Prajna Bhandary	6936	98,50	98,50	98,50	98,50

(Pham, Nguyen and Huh 2023)	2023	Study 12	Republic of Korea	Supervised	fmd	Deep Learning	YoloV5 small	3	D1- FMD Dataset by Larxel D2-FMLD D3- Custom Dataset(Composed of Kaggle and YouTube Videos)	89021	90,80	86,00	80,00	80,00
(Yu et al. 2023)	2023	Study 13	China	Supervised	fmd	Deep Learning	Yolov4 CSPDarkNet-53	1	Custom RFM Dataset	12133	73,82	78,02	76,67	77,00
(Bania 2023)	2023	Study 14	India	Supervised	fmd	Transfer Learning and Deep Learning	ResNet50 Inception-V3 VGG-16	2	D1-FMD Custom Dataset (Composed of multiple Kaggle Datasets) D2- FMD Dataset by Omkar Gurav	11376	99,97	99,97	99,97	99,97
(Habib et al. 2022)	2022	Study 15	Saudi Arabia	Supervised	fmd	Deep Learning	CNN MobileNetV2	3	D1- FMD Dataset by Omkar Gurav D2-FM by Oumina, Makhfi, and Hamdi D3-RMFR	103929	99,98	99,96	99,97	99,97
(Sethi, Kathuria and Kaushik 2021)	2021	Study 16	India	Supervised	fmd	Deep Learning	ResNet50 MobileNet AlexNet	1	Custom Dataset	25876	98,20	98,92	98,24	98,57
(Kumar et al. 2022)	2022	Study 17	India	Supervised	fmd	Deep Learning	Hybrid Deep CNN	3	D1-PyImage Search Reader Prajna Bhandary D2-Witkowski's Medical Mask Dataset D3-Custom Dataset	3440	99,07	98,13	98,91	98,52
(Kamil et al. 2023)	2023	Study 18	Malaysia	Supervised	fmd	Machine Learning	SVM PHP	2	Custom Datasets: D 1- Original Images D 2- Synthetic Masks Applied	600	80,00	100	80,00	88,89
(Loey et al. 2021)	2021	Study 19	USA	Supervised	fmd	Machine Learning and Deep Learning	ResNet50 SVM & Decision Trees & Ensemble Algorithm	4	D1- RMFD D2-SMFD D3- LFW D4-Custom Dataset (Composed of RMFD+SMFD)	29570	99,71	99,00	99,00	99,00
(Bhaik et al. 2021)	2022	Study 20	India	Supervised	fmd	Transfer Learning and Deep Learning	CNN Caffe Framework MobileNetV2-Transfer Learning	1	Custom Dataset	500	93,58	92,15	94,00	93,10
(Hussain et al. 2021)	2021	Study 21	Pakistan	Supervised	fmd	Deep Learning	VGG-16 CNN InceptionV3	3	D1-MAFA D2-Masked Face-Net D3-FMLD(Bing)	83565	99,81	99,00	98,00	98,00
(Özyurt, Mira and Çoban 2022)	2022	Study 22	Turkey	Supervised	fmd	Deep Learning	MobileNetV2	1	MaskedFace-Net	5774	97,00	97,00	96,00	97,00

(Xu et al. 2023)	2023	Study 23	China	Supervised	fmd	Deep Learning	YoloV4 YOLO-ARGhost	2	D1-AIZOO D2-FMDD by Wobot Intelligence	11343	99,34	95,10	94,30	95,00
(Kowalczyk, Sobotka and Rumiński 2023)	2023	Study 24	Poland	Semi-Supervised	fmd	Deep Learning	YoloV5	1	Face with Mask Thermal Custom Dataset	9394	91,00	95,00	94,00	94,49
(SÜNNETCİ et al. 2023)	2023	Study 25	Turkey	Supervised	fmd	Machine Learning and Deep Learning	GoogLeNet Linear SVM	1	Face Mask Classification Dataset by Dhruv Makwana	440	99,50	99,55	99,55	99,55
(Ottakath et al. 2022)	2022	Study 26	Qatar	Supervised	fmd	Transfer Learning and Deep Learning	YoloR	2	D1-ViDMASK(Custom Dataset made up of videos) D2-Moxa3K	30 600	92,40	84,00	91,70	87,68
(Yahya et al. 2021)	2021	Study 27	Malaysia	Supervised	fmd	Deep Learning	ResNet101	1	Custom Dataset	400	100	100	100	100
(Yu and Zhang 2021)	2021	Study 28	China	Supervised	fmd	Deep Learning	YOLOv4 CSPDarkNet53 PANet	2	D1-RMFD D2-MaskedFace-Net	10855	98,30	95,10	98,20	96,70
(Zhao, Zou, and Wu 2023)	2023	Study 29	China	Supervised	fmd	Deep Learning	YOLO-v4 CBAM	2	D1-MAFA D2-WiderFace	7959	93,56	95,15	95,64	95,39
(Goyal et al. 2022)	2022	Study 30	India	Supervised	fmd	Deep Learning	Custom CNN Streamlit	1	Custom Dataset (acquired from publicly available datasets i.e., Combination of datasets including the FMD Dataset by Omkar)	4000	98,00	98,00	98,00	98,00
(Głowacka and Rumiński 2021)	2021	Study 31	Poland	Supervised	fmd	Deep Learning	YoloV3	1	Custom dataset	7920	99,30	69,10	99,80	81,65
(Kumar et al. 2021)	2021	Study 32	India	Supervised	fmd	Deep Learning	Modified tiny YoloV4	3	D1-ImageNet, D2-MS COCO D3-PASCAL VOC	52635	71,69	78,00	79,00	78,00
(Sharma, Gautam, and Singh 2023)	2023	Study 33	India	Supervised	fmd	Deep Learning	YoloV5	1	Custom Novel Dataset (acquired from publicly available datasets i.e., Moxa3k benchmark, google and Bing images)	10 000	80,80	86,10	73,40	79,00
(Umer et al. 2023)	2023	Study 34	Pakistan	Supervised	fmd	Deep Learning	Custom CNN	3	D1-RILFD(Custom Dataset) D2-MAFA D3-MOXA	8750	97,50	96,20	97,34	96,77
(Chen et al. 2023)	2023	Study 35	China	Supervised	fmd	Transfer Learning and Deep Learning	YoloX	2	D1-WMD D2-PWMFD	17009	86,06	99,62	93,11	93,00
(Kumar, Kalia and Kalia 2022)	2022	Study 36	India	Supervised	fmd	Deep Learning	ETL-YOLO v4	1	D1- Face Mask Dataset by Kumar et al	52635	86,89	85,00	72,00	78,00
(Rafidison et al. 2023)	2023	Study 37	East Africa Madagascar	Supervised	fmd	Deep Learning	Pulse Couple Neural Network(PCNN) & Fully Connected Neural Network(FCNN)	6	D1-Internet source Dataset by data flair training. D2-AIZOO D3-Moxa3K D4-RFMD D5- Witkowski's Medical Mask Dataset(MMD)	17473	86,68	94,00	92,50	93,00

									D6- FMD Custom Dataset (Composed of multiple Kaggle Datasets)					
(Kumar et al. 2021)	2021	Study 38	India	Supervised	fmd	Deep Learning	Tiny YOLO v4 SPP CSPDarknet-53	1	Custom Dataset (Google & Bing) Images	52635	84,42	84,00	70,00	77,00
(Jayaswal and Dixit 2023)	2023	Study 39	India	Supervised	fmd	Deep Learning	InceptionV3 SSMD	3	D1-RFMD D2-RTFMD D3-RTFMD-V2	4801	97,00	97,52	97,34	97,43
(Farman et al. 2022)	2022	Study 40	Pakistan	Supervised	fmd	Deep Learning	Custom CNN developed	1	Custom Dataset (acquired from publicly available sources from the internet)	7740	99,00	98,00	98,00	98,00
(Dewi and Chen 2022)	2022	Study 41	Indonesia	Supervised	fmd	Deep Learning	Yolo V4 CSP SPP	2	D1-Medical Face Mask(MFMD) D2- Witkowski's Medical Mask (MMD)	1415	99,26	97,00	99,00	98,00
(Kumar and Bansal 2023)	2023	Study 42	India	Supervised	fmd	Deep Learning	Caffe-MobileNetV2 (CMNV2)	1	PyImage Search Reader Prajna Bhandary	1381	99,64	100	99,28	99,64
(Balasubaramanian et al. 2023)	2023	Study 43	Oman	Supervised	fmd	Deep Learning	Depth-wise Separable CNN using PCA & Stacked Autoencoder). MobileNet	1	Custom Dataset	1328	94,10	97,61	94,46	96,00
(Walia et al. 2021)	2021	Study 44	India	Supervised	fmd	Deep Learning	Stacked ResNet-50 YoloV5 DBSCAN	2	D1-RMFD D2-Custom Dataset(Google Images)	4494	87,00	71,00	92,00	80,14
(Nagrath et al. 2021)	2021	Study 45	India	Supervised	fmd	Deep Learning	SSD MobileNetV2	2	D1- Witkowski's Medical Mask Dataset(MMD) D2-Prajna Bhandary dataset	7575	92,64	94,00	93,00	93,00
(Said 2020)	2020	Study 46	Tunisia	Supervised	fmd	Deep Learning	CNN (light weight) Yolo Pynq Z(Platform)	4	D1-RMFD D2-MFDD by Wang et al D3-SMFRD D3-MAFA	685388	97,00	94,60	95,80	95,20
(Teboulbi et al. 2021)	2021	Study 47	Tunisia	Supervised	fmd	Deep Learning	VGG-19 CNN	1	Custom dataset	3835	100	99,00	99,00	99,00
(Sheikh and Zafar 2023)	2023	Study 48	India	Supervised	fmd	Deep Learning	MobileNetv2	1	Sophisticated Face Mask Dataset(Custom Dataset. Acquired from publicly available datasets i.e., MAFA and MFDD)	14535	95,83	95,00	94,00	94,00
(Benifa et al. 2023)	2023	Study 49	India	Supervised	fmd	Deep Learning	FMDNet YoloV5	5	D1- Pins Face Recognition	98145	99,00	98,00	99,00	99,00

									dataset D2-FMD Dataset by Omkar Gurav D3-LFW D4-RMFD D5-YOLO-Medical-Mask-Dataset					
(Marwa and Kais 2022)	2022	Study 50	Tunisia	Supervised	mfr	Machine Learning and Deep Learning	FaceNet SVM KNN	2	D1-RMFD D2-SMFD	595 000	91,07	89,25	90,50	89,65
(Wang, Li, and Zou 2023)	2023	Study 51	China	Supervised	mfr	Deep Learning	ConvNeXt-T(FaceNet with ECA)	3	D1-Custom Dataset D2-CASIA-WebFace D3-LFW	508952	99,48	99,56	98,50	99,70
(Pann and Lee 2022)	2022	Study 52	Republic of Korea	Supervised	mfr	Deep Learning	ArcFace CBAM CNN ReNet50	5	D1-CASIA-WebFace D2-LFW D3-AgeDB-30 D4-CFP-FP D5-MFR2	789296	97,45	96,54	98,32	97,42
(Cimmino et al. 2022)	2022	Study 53	Italy	Supervised	mfr	Machine Learning	LBP-TOP 68 Landmark detection algorithm GAN MLP	2	D1-M2FRED D3-XM2VTS	412800	85,60	73,40	85,60	79,03
(Al-Rammahi 2022)	2022	Study 54	Iraq	Supervised	fmd	Deep Learning	MobileNetV2 CNN	1	RMFD	3832	99,21	99,00	99,00	99,00
(Fazeli Ardekani, Tale and Parseh 2023)	2023	Study 55	Iran	Supervised	fmd	Deep Learning	Custom CNN	1	Face Mask Detection ~ 12 K Images	36000	99,02	99,00	99,00	99,00
(Chen et al. 2022)	2022	Study 56	China	Supervised	mfr	Deep Learning	ESRGAN EfficientNet RetinaFace	2	D1-RMFRD D2-SMFRD	595000	98,70	91,90	91,50	91,70
(Mahmoud, Alharbi and Alghamdi 2022)	2022	Study 57	Saudi Arabia	Supervised	fmd	Deep Learning	R-CNN CNN LHMD DS-Face	1	MaskedFace-Net	137016	91,00	94,50	57,50	71,50
(Yu et al. 2023)	2023	Study 58	China	Supervised	fmd	Deep Learning	YOLO-v5s ASFF BiFPN	4	D1-AIZOO D2- FDM(Open-source Internet Images) D3-RFMD D4-Custom Dataset	23656	89,50	89,50	84,50	86,93
(Peng et al. 2023)	2023	Study 59	China	Supervised	mfr	Deep Learning	Xception network SSD ResNet ML	3	All collected from Wuhan University: D1-MFDD D2-RMFRD D3-SMFRD	619771	98,30	97,90	96,10	97,40
(Manzoor et al. 2022)	2022	Study 60	South Korea	Supervised	fmd	Transfer Learning and Deep Learning	CNN Seven transfer learning based custom models: Inception-v3 VGG16	1	CelebFace	12000	94,28	97,00	92,00	94,00

							denseNet121 resNet50 NASNetMobile XceptionNet MobileNet-v2							
(Talahua et al. 2021)	2021	Study 61	Spain	Supervised	fmd	Deep Learning	MobileNetV2 OpenCV FaceNet	1	Custom Dataset	13359	99,65	100	100	100
(Talahua et al. 2021)	2021	Study 62	Spain	Supervised	mfr	Deep Learning	MobileNetV2 OpenCV FaceNet	1	Custom Dataset	13359	99,52	99,00	99,00	99,00
(Kumar et al. 2023)	2023	Study 63	India	Supervised	mfr	Deep Learning	YOLOv3 Darknet-53	1	Custom Dataset	900	98,73	95,00	97,00	96,00
(Mar-Cupido et al. 2022)	2022	Study 64	Mexico	Supervised	fmd	Deep Learning	ResNet101v2	5	D1- Face Mask Classification Dataset by Dhruv Makwana D2-FMD Dataset by Larxel D3-MFR2 D4-MAFA D5-Custom Dataset(Bing and Google Images)	48281	98,00	98,01	98,02	98,02
(Hu et al. 2023)	2023	Study 65	China	Supervised	fmd	Deep Learning	ECGYOLO YOLOv7tiny GhostNet ECA	2	D1-WIDER Face D2-MAFA	10043	92,75	96,20	87,60	91,70
(Akingbesote et al. 2023)	2023	Study 66	Lithuania	Supervised	mfr	Deep Learning	Pareto-optimized FaceNet GAN	2	D1-CASIA WebFace D2-VGG-Face	2517729	95,00	93,96	93,96	93,87
(Wang et al. 2023)	2023	Study 67	China	Supervised	fmd	Deep Learning	YOLO-GBC GAM SSP	1	Custom Dataset	11317	91,20	91,10	86,40	88,69
(Guo et al. 2022)	2022	Study 68	China	Supervised	fmd	Deep Learning	YOLOv5-CBD SPP FPN, PAN	3	D1-FMD Dataset by Omkar Gurav D2-AFLW D3-Custom Dataset(Web images using crawler)	8730	96,70	96,30	92,50	94,36
(Yang et al. 2022)	2021	Study 69	United Kingdom	Supervised	fmd	Transfer Learning and Deep Learning	Mask R-CNN ResNet-101 FPN	1	Mask Dataset(Kaggle)	2447	92,10	93,00	96,00	94,00
(Zia et al. 2022)	2022	Study 70	United Arab Emirates (UAE)	Supervised	fmd	Deep Learning	YOLO MobileNetV2	8	Custom Dataset 1 Custom Dataset 2 Custom Dataset 3 Custom Dataset 4 Custom Dataset 5 Custom Dataset 6 Custom Dataset 7 Custom Dataset 8	81400	95,00	91,40	65,40	76,24
(Hung 2023)	2023	Study 71	Taiwan	Supervised	fmd	Deep Learning	YOLO CNN	1	Custom Dataset	1000	90,00	90,00	93,00	91,00

(Rahmani et al. 2022)	2022	Study 72	Saudi Arabia	Supervised	fmd	Deep Learning	MobilNetV2	4	D1- COVID Dataset D2- FMD Dataset by Larxel D3-PyImage Search reader Prajna Bhandary D4- Witkowski's Medical Mask Dataset(MMD)	282911	99,00	99,00	99,00	99,00
(Asghar et al. 2022)	2022	Study 73	Malaysia	Supervised	fmd	Deep Learning	Depthwise Separable CNN MobileNet	2	D1-AIZOO D2-Moxa3K	11000	93,14	92,00	92,00	92,00
(Meivel *et al.* 2022)	2022	Study 74	India	Supervised	fmd	Deep Learning	faster R-CNN OpenCV YoloV3	1	Custom Dataset	26811	95,00	99,00	86,00	92,00
(Wang, Zhao, and Chen 2021)	2021	Study 75	China	Supervised	fmd	Transfer Learning and Deep Learning	Faster RCNN InceptionV2	1	WMD Dataset	9398	94,19	93,54	94,84	94,19

ANNEXURE B: PROPOSAL APPROVAL AND ETHICS CONSIDERATION

MR

Mogiveny Rajkoomar

To: C Pillay (21709836); Seena Joseph <SeenaJ@dut.ac.za>; **+1 other** Fri 2023/11/17 02:41

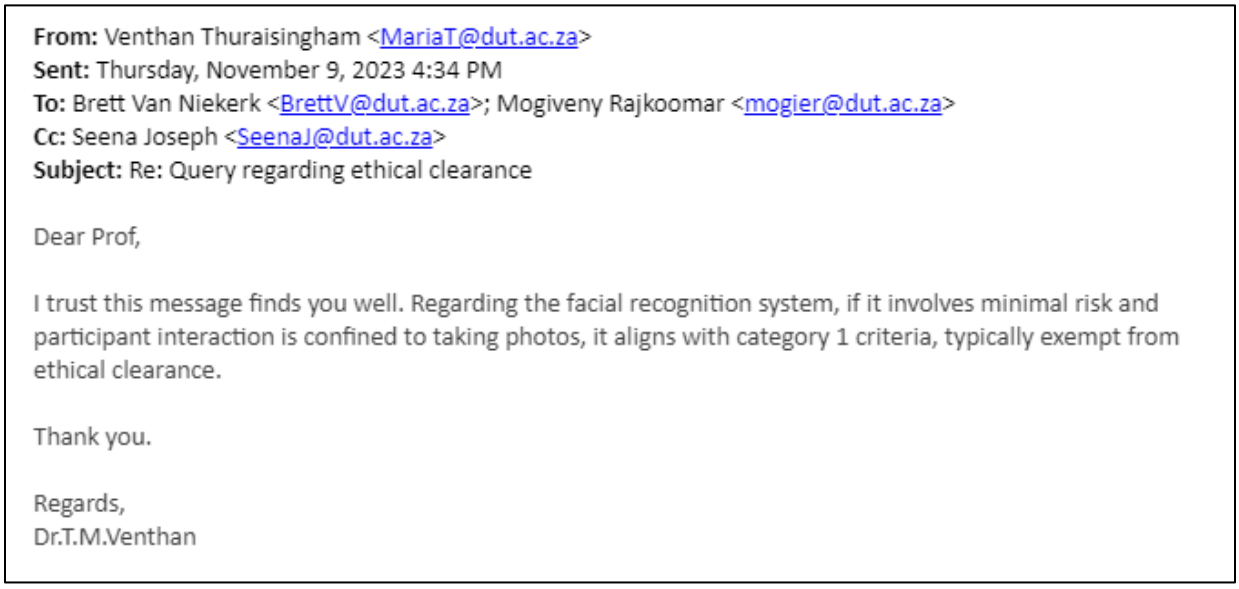
Cc: Sboniso Mjaja <SbonisoM3@dut.ac.za>

PG2A- C Pillay 21709836 17 ...
854 KB

Dear Ms Pillay
I trust that you are doing well.

The recommended changes arising from the FRC review have been adequately addressed. It is my pleasure to inform you that your Pg2a research proposal is now approved. I attach the FRC approved and signed Pg2a.

Wishing you all the best for the completion of your studies.
Kind regards
Dr Rajkoomar



From: Venthan Thuraisingham <MariaT@dut.ac.za>
Sent: Thursday, November 9, 2023 4:34 PM
To: Brett Van Niekerk <BrettV@dut.ac.za>; Mogiveny Rajkoomar <mogier@dut.ac.za>
Cc: Seena Joseph <SeenaJ@dut.ac.za>
Subject: Re: Query regarding ethical clearance

Dear Prof,

I trust this message finds you well. Regarding the facial recognition system, if it involves minimal risk and participant interaction is confined to taking photos, it aligns with category 1 criteria, typically exempt from ethical clearance.

Thank you.

Regards,
Dr.T.M.Venthan

ANNEXURE C: ANALYSIS OF MODEL TRAINING DYNAMICS ACROSS EPOCHS

```
Epoch 1/35
150/150 [==============================] - 217s 1s/step - loss: 0.8164 - accuracy: 0.5600 - val_loss: 0.5753 - val_accuracy:
0.5502 - lr: 1.0000e-04
Epoch 2/35
150/150 [==============================] - 195s 1s/step - loss: 0.6299 - accuracy: 0.6998 - val_loss: 0.4506 - val_accuracy:
0.7575 - lr: 9.5000e-05
Epoch 3/35
150/150 [==============================] - 191s 1s/step - loss: 0.4343 - accuracy: 0.8361 - val_loss: 0.2788 - val_accuracy:
0.8763 - lr: 9.0250e-05
Epoch 4/35
150/150 [==============================] - 176s 1s/step - loss: 0.3016 - accuracy: 0.9072 - val_loss: 0.2242 - val_accuracy:
0.9030 - lr: 8.5737e-05
Epoch 5/35
150/150 [==============================] - 169s 1s/step - loss: 0.2106 - accuracy: 0.9475 - val_loss: 0.1637 - val_accuracy:
0.9348 - lr: 8.1451e-05
Epoch 6/35
150/150 [==============================] - 180s 1s/step - loss: 0.1732 - accuracy: 0.9586 - val_loss: 0.1478 - val_accuracy:
0.9398 - lr: 7.7378e-05
Epoch 7/35
150/150 [==============================] - 171s 1s/step - loss: 0.1383 - accuracy: 0.9653 - val_loss: 0.0823 - val_accuracy:
0.9682 - lr: 7.3509e-05
Epoch 8/35
150/150 [==============================] - 190s 1s/step - loss: 0.1138 - accuracy: 0.9718 - val_loss: 0.0824 - val_accuracy:
0.9716 - lr: 6.9834e-05
Epoch 9/35
150/150 [==============================] - 176s 1s/step - loss: 0.0976 - accuracy: 0.9755 - val_loss: 0.0568 - val_accuracy:
0.9766 - lr: 6.6342e-05
Epoch 10/35
150/150 [==============================] - 157s 1s/step - loss: 0.0832 - accuracy: 0.9814 - val_loss: 0.1106 - val_accuracy:
0.9582 - lr: 6.3025e-05
Epoch 11/35
150/150 [==============================] - 156s 1s/step - loss: 0.0842 - accuracy: 0.9808 - val_loss: 0.1160 - val_accuracy:
0.9582 - lr: 5.9874e-05
Epoch 12/35
150/150 [==============================] - 155s 1s/step - loss: 0.0735 - accuracy: 0.9839 - val_loss: 0.0615 - val_accuracy:
0.9732 - lr: 5.6880e-05
Epoch 13/35
150/150 [==============================] - 177s 1s/step - loss: 0.0737 - accuracy: 0.9801 - val_loss: 0.0403 - val_accuracy:
0.9833 - lr: 5.4036e-05
Epoch 14/35
150/150 [==============================] - 194s 1s/step - loss: 0.0636 - accuracy: 0.9845 - val_loss: 0.0342 - val_accuracy:
0.9866 - lr: 5.1334e-05
Epoch 15/35
150/150 [==============================] - 185s 1s/step - loss: 0.0728 - accuracy: 0.9827 - val_loss: 0.0202 - val_accuracy:
0.9916 - lr: 4.8767e-05
Epoch 16/35
150/150 [==============================] - 169s 1s/step - loss: 0.0577 - accuracy: 0.9831 - val_loss: 0.0127 - val_accuracy:
0.9950 - lr: 4.6329e-05
Epoch 17/35
150/150 [==============================] - 182s 1s/step - loss: 0.0517 - accuracy: 0.9868 - val_loss: 0.0079 - val_accuracy:
0.9967 - lr: 4.4013e-05
Epoch 18/35
150/150 [==============================] - 159s 1s/step - loss: 0.0488 - accuracy: 0.9893 - val_loss: 0.0072 - val_accuracy:
0.9983 - lr: 4.1812e-05
Epoch 19/35
150/150 [==============================] - 161s 1s/step - loss: 0.0525 - accuracy: 0.9854 - val_loss: 0.0128 - val_accuracy:
0.9933 - lr: 3.9721e-05
```

```
Epoch 20/35
150/150 [==============================] - 182s 1s/step - loss: 0.0532 - accuracy: 0.9849 - val_loss: 0.0135 - val_accuracy:
0.9933 - lr: 3.7735e-05
Epoch 21/35
150/150 [==============================] - 182s 1s/step - loss: 0.0393 - accuracy: 0.9912 - val_loss: 0.0091 - val_accuracy:
0.9950 - lr: 3.5849e-05
Epoch 22/35
150/150 [==============================] - 194s 1s/step - loss: 0.0355 - accuracy: 0.9918 - val_loss: 0.0085 - val_accuracy:
0.9950 - lr: 3.4056e-05
Epoch 23/35
150/150 [==============================] - 179s 1s/step - loss: 0.0432 - accuracy: 0.9893 - val_loss: 0.0056 - val_accuracy:
0.9983 - lr: 3.2353e-05
Epoch 24/35
150/150 [==============================] - 173s 1s/step - loss: 0.0394 - accuracy: 0.9906 - val_loss: 0.0053 - val_accuracy:
0.9983 - lr: 3.0736e-05
Epoch 25/35
150/150 [==============================] - 168s 1s/step - loss: 0.0380 - accuracy: 0.9895 - val_loss: 0.0042 - val_accuracy:
1.0000 - lr: 2.9199e-05
Epoch 26/35
150/150 [==============================] - 165s 1s/step - loss: 0.0361 - accuracy: 0.9904 - val_loss: 0.0054 - val_accuracy:
0.9967 - lr: 2.7739e-05
Epoch 27/35
150/150 [==============================] - 166s 1s/step - loss: 0.0376 - accuracy: 0.9895 - val_loss: 0.0113 - val_accuracy:
0.9933 - lr: 2.6352e-05
Epoch 28/35
150/150 [==============================] - 164s 1s/step - loss: 0.0445 - accuracy: 0.9883 - val_loss: 0.0033 - val_accuracy:
1.0000 - lr: 2.5034e-05
Epoch 29/35
150/150 [==============================] - 162s 1s/step - loss: 0.0316 - accuracy: 0.9916 - val_loss: 0.0024 - val_accuracy:
1.0000 - lr: 2.3783e-05
Epoch 30/35
150/150 [==============================] - 160s 1s/step - loss: 0.0362 - accuracy: 0.9910 - val_loss: 0.0034 - val_accuracy:
0.9983 - lr: 2.2594e-05
Epoch 31/35
150/150 [==============================] - 177s 1s/step - loss: 0.0381 - accuracy: 0.9908 - val_loss: 0.0026 - val_accuracy:
1.0000 - lr: 2.1464e-05
Epoch 32/35
150/150 [==============================] - 175s 1s/step - loss: 0.0294 - accuracy: 0.9927 - val_loss: 0.0041 - val_accuracy:
1.0000 - lr: 2.0391e-05
Epoch 33/35
150/150 [==============================] - 173s 1s/step - loss: 0.0259 - accuracy: 0.9929 - val_loss: 0.0027 - val_accuracy:
1.0000 - lr: 1.9371e-05
Epoch 34/35
150/150 [==============================] - 175s 1s/step - loss: 0.0280 - accuracy: 0.9939 - val_loss: 0.0034 - val_accuracy:
1.0000 - lr: 1.8403e-05
Epoch 35/35
150/150 [==============================] - 188s 1s/step - loss: 0.0214 - accuracy: 0.9944 - val_loss: 0.0024 - val_accuracy:
1.0000 - lr: 1.7482e-05
19/19 [==============================] - 4s 189ms/step - loss: 0.0012 - accuracy: 1.0000
Test loss: 0.0012433170340955257
Test accuracy: 1.0
```

Figure 1: Training the Model on the 'face mask detection' Dataset

```
Epoch 1/35
125/125 [==============================] - 148s 1s/step - loss: 0.8853 - accuracy: 0.5360 - val_loss: 0.5504 - val_accuracy:
0.6900 - lr: 1.0000e-04
Epoch 2/35
125/125 [==============================] - 136s 1s/step - loss: 0.7094 - accuracy: 0.6543 - val_loss: 0.3975 - val_accuracy:
0.7820 - lr: 9.5000e-05
Epoch 3/35
125/125 [==============================] - 134s 1s/step - loss: 0.5220 - accuracy: 0.7922 - val_loss: 0.3148 - val_accuracy:
0.8620 - lr: 9.0250e-05
Epoch 4/35
125/125 [==============================] - 136s 1s/step - loss: 0.3875 - accuracy: 0.8737 - val_loss: 0.2308 - val_accuracy:
0.9240 - lr: 8.5737e-05
Epoch 5/35
125/125 [==============================] - 149s 1s/step - loss: 0.2871 - accuracy: 0.9190 - val_loss: 0.2091 - val_accuracy:
0.9340 - lr: 8.1451e-05
Epoch 6/35
125/125 [==============================] - 143s 1s/step - loss: 0.2182 - accuracy: 0.9475 - val_loss: 0.1420 - val_accuracy:
0.9480 - lr: 7.7378e-05
Epoch 7/35
125/125 [==============================] - 138s 1s/step - loss: 0.1709 - accuracy: 0.9615 - val_loss: 0.1041 - val_accuracy:
0.9640 - lr: 7.3509e-05
Epoch 8/35
125/125 [==============================] - 134s 1s/step - loss: 0.1549 - accuracy: 0.9645 - val_loss: 0.0672 - val_accuracy:
0.9760 - lr: 6.9834e-05
Epoch 9/35
125/125 [==============================] - 132s 1s/step - loss: 0.1303 - accuracy: 0.9707 - val_loss: 0.0639 - val_accuracy:
0.9760 - lr: 6.6342e-05
Epoch 10/35
125/125 [==============================] - 131s 1s/step - loss: 0.1212 - accuracy: 0.9707 - val_loss: 0.0460 - val_accuracy:
0.9880 - lr: 6.3025e-05
Epoch 11/35
125/125 [==============================] - 131s 1s/step - loss: 0.1093 - accuracy: 0.9720 - val_loss: 0.0367 - val_accuracy:
0.9920 - lr: 5.9874e-05
Epoch 12/35
125/125 [==============================] - 128s 1s/step - loss: 0.0927 - accuracy: 0.9785 - val_loss: 0.0466 - val_accuracy:
0.9840 - lr: 5.6880e-05
Epoch 13/35
125/125 [==============================] - 140s 1s/step - loss: 0.0813 - accuracy: 0.9810 - val_loss: 0.0250 - val_accuracy:
0.9960 - lr: 5.4036e-05
Epoch 14/35
125/125 [==============================] - 139s 1s/step - loss: 0.0790 - accuracy: 0.9822 - val_loss: 0.0177 - val_accuracy:
0.9960 - lr: 5.1334e-05
Epoch 15/35
125/125 [==============================] - 136s 1s/step - loss: 0.0823 - accuracy: 0.9793 - val_loss: 0.0170 - val_accuracy:
0.9980 - lr: 4.8767e-05
Epoch 16/35
125/125 [==============================] - 130s 1s/step - loss: 0.0656 - accuracy: 0.9877 - val_loss: 0.0101 - val_accuracy:
0.9980 - lr: 4.6329e-05
Epoch 17/35
125/125 [==============================] - 135s 1s/step - loss: 0.0683 - accuracy: 0.9843 - val_loss: 0.0089 - val_accuracy:
0.9980 - lr: 4.4013e-05
Epoch 18/35
125/125 [==============================] - 131s 1s/step - loss: 0.0596 - accuracy: 0.9872 - val_loss: 0.0172 - val_accuracy:
0.9980 - lr: 4.1812e-05
Epoch 19/35
125/125 [==============================] - 142s 1s/step - loss: 0.0473 - accuracy: 0.9912 - val_loss: 0.0120 - val_accuracy:
0.9980 - lr: 3.9721e-05
Epoch 20/35
125/125 [==============================] - 138s 1s/step - loss: 0.0392 - accuracy: 0.9920 - val_loss: 0.0095 - val_accuracy:
0.9980 - lr: 3.7735e-05
```

```
Epoch 21/35
125/125 [==============================] - 133s 1s/step - loss: 0.0510 - accuracy: 0.9887 - val_loss: 0.0118 - val_accuracy:
0.9980 - lr: 3.5849e-05
Epoch 22/35
125/125 [==============================] - 133s 1s/step - loss: 0.0593 - accuracy: 0.9872 - val_loss: 0.0076 - val_accuracy:
0.9980 - lr: 3.4056e-05
Epoch 23/35
125/125 [==============================] - 135s 1s/step - loss: 0.0395 - accuracy: 0.9900 - val_loss: 0.0059 - val_accuracy:
0.9980 - lr: 3.2353e-05
Epoch 24/35
125/125 [==============================] - 138s 1s/step - loss: 0.0452 - accuracy: 0.9875 - val_loss: 0.0080 - val_accuracy:
0.9980 - lr: 3.0736e-05
Epoch 25/35
125/125 [==============================] - 137s 1s/step - loss: 0.0360 - accuracy: 0.9925 - val_loss: 0.0062 - val_accuracy:
0.9980 - lr: 2.9199e-05
Epoch 26/35
125/125 [==============================] - 132s 1s/step - loss: 0.0424 - accuracy: 0.9910 - val_loss: 0.0033 - val_accuracy:
1.0000 - lr: 2.7739e-05
Epoch 27/35
125/125 [==============================] - 135s 1s/step - loss: 0.0344 - accuracy: 0.9925 - val_loss: 0.0020 - val_accuracy:
1.0000 - lr: 2.6352e-05
Epoch 28/35
125/125 [==============================] - 139s 1s/step - loss: 0.0412 - accuracy: 0.9902 - val_loss: 0.0022 - val_accuracy:
1.0000 - lr: 2.5034e-05
Epoch 29/35
125/125 [==============================] - 145s 1s/step - loss: 0.0385 - accuracy: 0.9918 - val_loss: 0.0019 - val_accuracy:
1.0000 - lr: 2.3783e-05
Epoch 30/35
125/125 [==============================] - 137s 1s/step - loss: 0.0420 - accuracy: 0.9910 - val_loss: 0.0020 - val_accuracy:
1.0000 - lr: 2.2594e-05
Epoch 31/35
125/125 [==============================] - 163s 1s/step - loss: 0.0482 - accuracy: 0.9898 - val_loss: 0.0021 - val_accuracy:
1.0000 - lr: 2.1464e-05
Epoch 32/35
125/125 [==============================] - 138s 1s/step - loss: 0.0311 - accuracy: 0.9923 - val_loss: 0.0017 - val_accuracy:
1.0000 - lr: 2.0391e-05
Epoch 33/35
125/125 [==============================] - 157s 1s/step - loss: 0.0344 - accuracy: 0.9923 - val_loss: 0.0027 - val_accuracy:
1.0000 - lr: 1.9371e-05
Epoch 34/35
125/125 [==============================] - 138s 1s/step - loss: 0.0206 - accuracy: 0.9962 - val_loss: 0.0017 - val_accuracy:
1.0000 - lr: 1.8403e-05
Epoch 35/35
125/125 [==============================] - 134s 1s/step - loss: 0.0260 - accuracy: 0.9945 - val_loss: 0.0020 - val_accuracy:
1.0000 - lr: 1.7482e-05
16/16 [==============================] - 3s 184ms/step - loss: 0.0014 - accuracy: 1.0000
Test loss: 0.001355909975245595
Test accuracy: 1.0
```

Figure 2: Training the Model on the 'Face Mask Detection 12K images' Dataset

```
Epoch 1/35
63/63 [==============================] - 54s 841ms/step - loss: 1.2146 - accuracy: 0.7444 - val_loss: 2.6375 - val_accuracy:
0.9153 - lr: 9.5123e-05
Epoch 2/35
63/63 [==============================] - 50s 804ms/step - loss: 0.3523 - accuracy: 0.9342 - val_loss: 2.4150 - val_accuracy:
0.9512 - lr: 9.0484e-05
Epoch 3/35
63/63 [==============================] - 38s 602ms/step - loss: 0.2409 - accuracy: 0.9556 - val_loss: 2.1222 - val_accuracy:
0.9651 - lr: 8.6071e-05
Epoch 4/35
63/63 [==============================] - 38s 597ms/step - loss: 0.1803 - accuracy: 0.9711 - val_loss: 1.7503 - val_accuracy:
0.9721 - lr: 8.1873e-05
Epoch 5/35
63/63 [==============================] - 39s 623ms/step - loss: 0.1448 - accuracy: 0.9763 - val_loss: 1.3349 - val_accuracy:
0.9761 - lr: 7.7880e-05
Epoch 6/35
63/63 [==============================] - 38s 599ms/step - loss: 0.1198 - accuracy: 0.9822 - val_loss: 0.8783 - val_accuracy:
0.9751 - lr: 7.4082e-05
Epoch 7/35
63/63 [==============================] - 38s 610ms/step - loss: 0.0988 - accuracy: 0.9852 - val_loss: 0.4994 - val_accuracy:
0.9801 - lr: 7.0469e-05
Epoch 8/35
63/63 [==============================] - 40s 634ms/step - loss: 0.0831 - accuracy: 0.9892 - val_loss: 0.2730 - val_accuracy:
0.9851 - lr: 6.7032e-05
Epoch 9/35
63/63 [==============================] - 41s 649ms/step - loss: 0.0761 - accuracy: 0.9904 - val_loss: 0.1667 - val_accuracy:
0.9861 - lr: 6.3763e-05
Epoch 10/35
63/63 [==============================] - 41s 649ms/step - loss: 0.0654 - accuracy: 0.9918 - val_loss: 0.1120 - val_accuracy:
0.9861 - lr: 6.0653e-05
Epoch 11/35
63/63 [==============================] - 43s 691ms/step - loss: 0.0558 - accuracy: 0.9943 - val_loss: 0.0815 - val_accuracy:
0.9890 - lr: 5.7695e-05
Epoch 12/35
63/63 [==============================] - 42s 665ms/step - loss: 0.0538 - accuracy: 0.9944 - val_loss: 0.0704 - val_accuracy:
0.9861 - lr: 5.4881e-05
Epoch 13/35
63/63 [==============================] - 43s 678ms/step - loss: 0.0471 - accuracy: 0.9958 - val_loss: 0.0649 - val_accuracy:
0.9910 - lr: 5.2205e-05
Epoch 14/35
63/63 [==============================] - 51s 807ms/step - loss: 0.0426 - accuracy: 0.9965 - val_loss: 0.0599 - val_accuracy:
0.9890 - lr: 4.9659e-05
Epoch 15/35
63/63 [==============================] - 43s 685ms/step - loss: 0.0387 - accuracy: 0.9968 - val_loss: 0.0583 - val_accuracy:
0.9871 - lr: 4.7237e-05
Epoch 16/35
63/63 [==============================] - 44s 697ms/step - loss: 0.0332 - accuracy: 0.9985 - val_loss: 0.0556 - val_accuracy:
0.9910 - lr: 4.4933e-05
Epoch 17/35
63/63 [==============================] - 50s 802ms/step - loss: 0.0333 - accuracy: 0.9979 - val_loss: 0.0550 - val_accuracy:
0.9910 - lr: 4.2742e-05
Epoch 18/35
63/63 [==============================] - 41s 647ms/step - loss: 0.0296 - accuracy: 0.9983 - val_loss: 0.0492 - val_accuracy:
0.9910 - lr: 4.0657e-05
Epoch 19/35
63/63 [==============================] - 41s 658ms/step - loss: 0.0281 - accuracy: 0.9990 - val_loss: 0.0497 - val_accuracy:
0.9900 - lr: 3.8674e-05
Epoch 20/35
63/63 [==============================] - 43s 679ms/step - loss: 0.0251 - accuracy: 0.9994 - val_loss: 0.0494 - val_accuracy:
0.9910 - lr: 3.6788e-05
```

```
Epoch 21/35
63/63 [==============================] - 42s 669ms/step - loss: 0.0236 - accuracy: 0.9993 - val_loss: 0.0481 - val_accuracy:
0.9890 - lr: 3.4994e-05
Epoch 22/35
63/63 [==============================] - 40s 642ms/step - loss: 0.0242 - accuracy: 0.9986 - val_loss: 0.0451 - val_accuracy:
0.9910 - lr: 3.3287e-05
Epoch 23/35
63/63 [==============================] - 41s 655ms/step - loss: 0.0210 - accuracy: 0.9995 - val_loss: 0.0443 - val_accuracy:
0.9910 - lr: 3.1664e-05
Epoch 24/35
63/63 [==============================] - 40s 639ms/step - loss: 0.0214 - accuracy: 0.9991 - val_loss: 0.0444 - val_accuracy:
0.9910 - lr: 3.0119e-05
Epoch 25/35
63/63 [==============================] - 39s 624ms/step - loss: 0.0205 - accuracy: 0.9996 - val_loss: 0.0430 - val_accuracy:
0.9910 - lr: 2.8651e-05
Epoch 26/35
63/63 [==============================] - 48s 771ms/step - loss: 0.0198 - accuracy: 0.9990 - val_loss: 0.0414 - val_accuracy:
0.9930 - lr: 2.7253e-05
Epoch 27/35
63/63 [==============================] - 42s 670ms/step - loss: 0.0178 - accuracy: 0.9994 - val_loss: 0.0423 - val_accuracy:
0.9910 - lr: 2.5924e-05
Epoch 28/35
63/63 [==============================] - 42s 672ms/step - loss: 0.0174 - accuracy: 0.9999 - val_loss: 0.0399 - val_accuracy:
0.9940 - lr: 2.4660e-05
Epoch 29/35
63/63 [==============================] - 42s 666ms/step - loss: 0.0164 - accuracy: 0.9996 - val_loss: 0.0417 - val_accuracy:
0.9920 - lr: 2.3457e-05
Epoch 30/35
63/63 [==============================] - 41s 650ms/step - loss: 0.0162 - accuracy: 0.9996 - val_loss: 0.0405 - val_accuracy:
0.9930 - lr: 2.2313e-05
Epoch 31/35
63/63 [==============================] - 42s 662ms/step - loss: 0.0157 - accuracy: 0.9998 - val_loss: 0.0402 - val_accuracy:
0.9930 - lr: 2.1225e-05
Epoch 32/35
63/63 [==============================] - 48s 766ms/step - loss: 0.0153 - accuracy: 0.9998 - val_loss: 0.0387 - val_accuracy:
0.9920 - lr: 2.0190e-05
Epoch 33/35
63/63 [==============================] - 56s 885ms/step - loss: 0.0148 - accuracy: 0.9995 - val_loss: 0.0387 - val_accuracy:
0.9900 - lr: 1.9205e-05
Epoch 34/35
63/63 [==============================] - 50s 804ms/step - loss: 0.0139 - accuracy: 0.9998 - val_loss: 0.0395 - val_accuracy:
0.9910 - lr: 1.8268e-05
Epoch 35/35
63/63 [==============================] - 45s 708ms/step - loss: 0.0142 - accuracy: 1.0000 - val_loss: 0.0380 - val_accuracy:
0.9930 - lr: 1.7377e-05
32/32 [==============================] - 2s 49ms/step - loss: 0.0305 - accuracy: 0.9940
Test Accuracy: 0.9940
Test Loss: 0.0305
```

Figure 3: Training the Model on the 'MFR Dataset'

```
Epoch 1/35
69/69 [==============================] - 43s 604ms/step - loss: 2.0658 - accuracy: 0.6276 - val_loss: 3.6029 - val_accuracy:
0.9382 - lr: 9.5123e-05
Epoch 2/35
69/69 [==============================] - 46s 660ms/step - loss: 0.6213 - accuracy: 0.9315 - val_loss: 3.2755 - val_accuracy:
0.9782 - lr: 9.0484e-05
Epoch 3/35
69/69 [==============================] - 45s 654ms/step - loss: 0.3745 - accuracy: 0.9643 - val_loss: 2.8751 - val_accuracy:
0.9873 - lr: 8.6071e-05
Epoch 4/35
69/69 [==============================] - 48s 697ms/step - loss: 0.2669 - accuracy: 0.9753 - val_loss: 2.3197 - val_accuracy:
0.9882 - lr: 8.1873e-05
Epoch 5/35
69/69 [==============================] - 47s 683ms/step - loss: 0.2051 - accuracy: 0.9848 - val_loss: 1.6624 - val_accuracy:
0.9891 - lr: 7.7880e-05
Epoch 6/35
69/69 [==============================] - 46s 663ms/step - loss: 0.1669 - accuracy: 0.9870 - val_loss: 0.9781 - val_accuracy:
0.9927 - lr: 7.4082e-05
Epoch 7/35
69/69 [==============================] - 50s 722ms/step - loss: 0.1458 - accuracy: 0.9898 - val_loss: 0.4963 - val_accuracy:
0.9936 - lr: 7.0469e-05
Epoch 8/35
69/69 [==============================] - 48s 692ms/step - loss: 0.1240 - accuracy: 0.9907 - val_loss: 0.2565 - val_accuracy:
0.9945 - lr: 6.7032e-05
Epoch 9/35
69/69 [==============================] - 46s 674ms/step - loss: 0.1037 - accuracy: 0.9937 - val_loss: 0.1353 - val_accuracy:
0.9936 - lr: 6.3763e-05
Epoch 10/35
69/69 [==============================] - 46s 662ms/step - loss: 0.0887 - accuracy: 0.9962 - val_loss: 0.0921 - val_accuracy:
0.9936 - lr: 6.0653e-05
Epoch 11/35
69/69 [==============================] - 46s 665ms/step - loss: 0.0803 - accuracy: 0.9960 - val_loss: 0.0755 - val_accuracy:
0.9955 - lr: 5.7695e-05
Epoch 12/35
69/69 [==============================] - 46s 669ms/step - loss: 0.0740 - accuracy: 0.9962 - val_loss: 0.0635 - val_accuracy:
0.9936 - lr: 5.4881e-05
Epoch 13/35
69/69 [==============================] - 46s 665ms/step - loss: 0.0662 - accuracy: 0.9977 - val_loss: 0.0594 - val_accuracy:
0.9964 - lr: 5.2205e-05
Epoch 14/35
69/69 [==============================] - 49s 708ms/step - loss: 0.0605 - accuracy: 0.9970 - val_loss: 0.0499 - val_accuracy:
0.9945 - lr: 4.9659e-05
Epoch 15/35
69/69 [==============================] - 57s 829ms/step - loss: 0.0536 - accuracy: 0.9984 - val_loss: 0.0467 - val_accuracy:
0.9955 - lr: 4.7237e-05
Epoch 16/35
69/69 [==============================] - 55s 801ms/step - loss: 0.0518 - accuracy: 0.9981 - val_loss: 0.0450 - val_accuracy:
0.9955 - lr: 4.4933e-05
Epoch 17/35
69/69 [==============================] - 49s 700ms/step - loss: 0.0460 - accuracy: 0.9990 - val_loss: 0.0428 - val_accuracy:
0.9964 - lr: 4.2742e-05
Epoch 18/35
69/69 [==============================] - 45s 657ms/step - loss: 0.0434 - accuracy: 0.9989 - val_loss: 0.0400 - val_accuracy:
0.9973 - lr: 4.0657e-05
Epoch 19/35
69/69 [==============================] - 48s 702ms/step - loss: 0.0392 - accuracy: 0.9997 - val_loss: 0.0381 - val_accuracy:
0.9973 - lr: 3.8674e-05
```

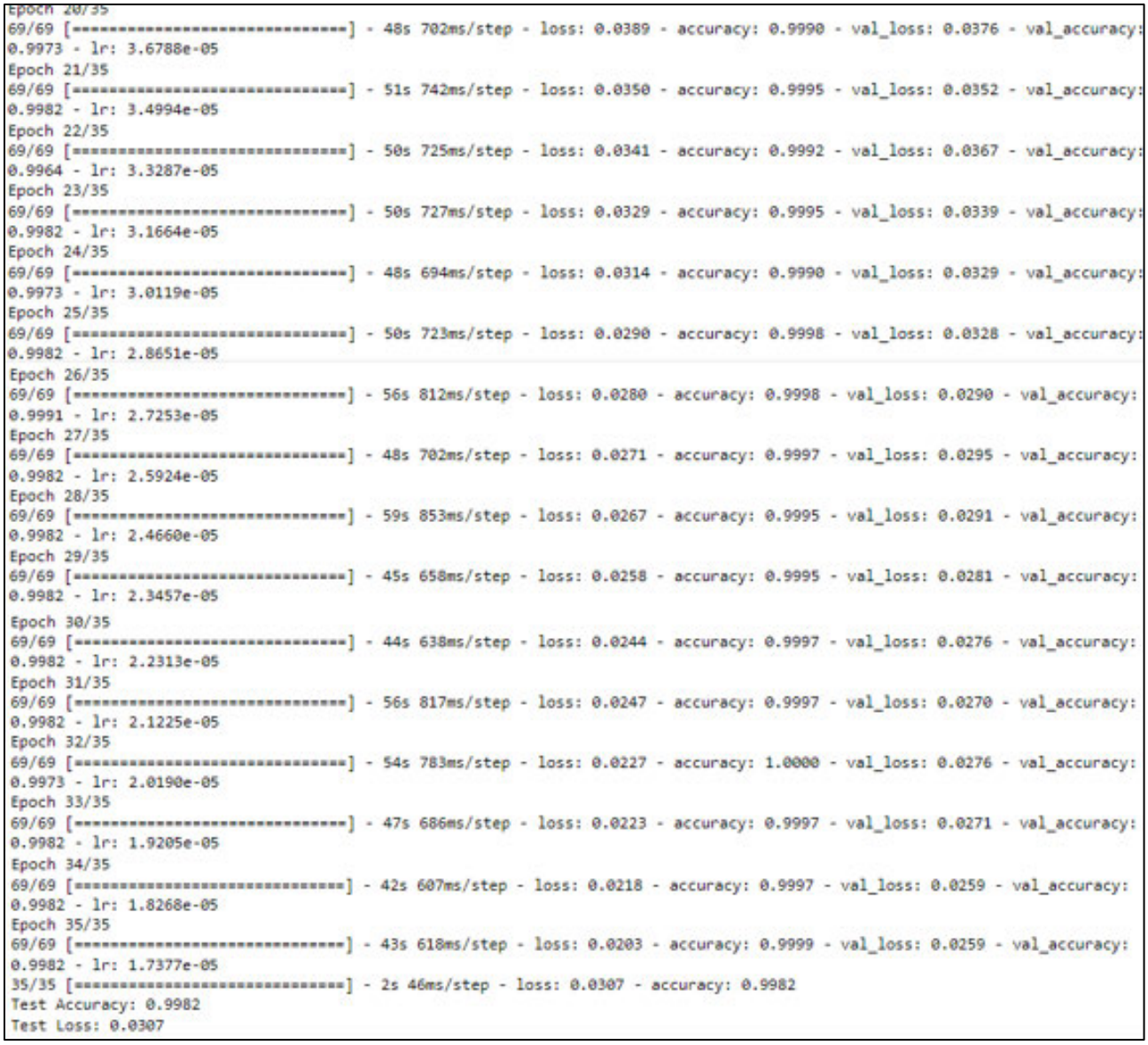


Figure 4: Training the Model on the 'MDMFR Dataset'

```
Epoch 1/35
35/35 [==============================] - 34s 886ms/step - loss: 0.8622 - accuracy: 0.7998 - val_loss: 2.0724 - val_accuracy:
0.9455 - lr: 9.5123e-05
Epoch 2/35
35/35 [==============================] - 31s 893ms/step - loss: 0.1681 - accuracy: 0.9677 - val_loss: 1.9441 - val_accuracy:
0.9691 - lr: 9.0484e-05
Epoch 3/35
35/35 [==============================] - 31s 875ms/step - loss: 0.0986 - accuracy: 0.9866 - val_loss: 1.8327 - val_accuracy:
0.9800 - lr: 8.6071e-05
Epoch 4/35
35/35 [==============================] - 30s 844ms/step - loss: 0.0706 - accuracy: 0.9893 - val_loss: 1.7070 - val_accuracy:
0.9909 - lr: 8.1873e-05
Epoch 5/35
35/35 [==============================] - 29s 842ms/step - loss: 0.0564 - accuracy: 0.9905 - val_loss: 1.5637 - val_accuracy:
0.9927 - lr: 7.7880e-05
Epoch 6/35
35/35 [==============================] - 29s 842ms/step - loss: 0.0430 - accuracy: 0.9959 - val_loss: 1.4063 - val_accuracy:
0.9945 - lr: 7.4082e-05
Epoch 7/35
35/35 [==============================] - 28s 812ms/step - loss: 0.0342 - accuracy: 0.9950 - val_loss: 1.2353 - val_accuracy:
0.9982 - lr: 7.0469e-05
Epoch 8/35
35/35 [==============================] - 29s 816ms/step - loss: 0.0287 - accuracy: 0.9980 - val_loss: 1.0238 - val_accuracy:
0.9964 - lr: 6.7032e-05
Epoch 9/35
35/35 [==============================] - 29s 831ms/step - loss: 0.0263 - accuracy: 0.9975 - val_loss: 0.8370 - val_accuracy:
0.9982 - lr: 6.3763e-05
Epoch 10/35
35/35 [==============================] - 28s 811ms/step - loss: 0.0223 - accuracy: 0.9984 - val_loss: 0.6478 - val_accuracy:
0.9964 - lr: 6.0653e-05
Epoch 11/35
35/35 [==============================] - 33s 932ms/step - loss: 0.0197 - accuracy: 0.9991 - val_loss: 0.4832 - val_accuracy:
0.9982 - lr: 5.7695e-05
Epoch 12/35
35/35 [==============================] - 29s 836ms/step - loss: 0.0189 - accuracy: 0.9977 - val_loss: 0.3395 - val_accuracy:
0.9964 - lr: 5.4881e-05
Epoch 13/35
35/35 [==============================] - 29s 841ms/step - loss: 0.0160 - accuracy: 0.9995 - val_loss: 0.2230 - val_accuracy:
0.9964 - lr: 5.2205e-05
Epoch 14/35
35/35 [==============================] - 29s 826ms/step - loss: 0.0138 - accuracy: 1.0000 - val_loss: 0.1410 - val_accuracy:
0.9982 - lr: 4.9659e-05
Epoch 15/35
35/35 [==============================] - 29s 839ms/step - loss: 0.0125 - accuracy: 0.9993 - val_loss: 0.0921 - val_accuracy:
0.9982 - lr: 4.7237e-05
Epoch 16/35
35/35 [==============================] - 29s 832ms/step - loss: 0.0117 - accuracy: 0.9998 - val_loss: 0.0617 - val_accuracy:
0.9982 - lr: 4.4933e-05
Epoch 17/35
35/35 [==============================] - 29s 828ms/step - loss: 0.0108 - accuracy: 0.9995 - val_loss: 0.0435 - val_accuracy:
0.9982 - lr: 4.2742e-05
Epoch 18/35
35/35 [==============================] - 31s 868ms/step - loss: 0.0098 - accuracy: 0.9995 - val_loss: 0.0313 - val_accuracy:
0.9982 - lr: 4.0657e-05
Epoch 19/35
35/35 [==============================] - 30s 853ms/step - loss: 0.0092 - accuracy: 1.0000 - val_loss: 0.0233 - val_accuracy:
0.9982 - lr: 3.8674e-05
```

```
Epoch 20/35
35/35 [==============================] - 33s 954ms/step - loss: 0.0089 - accuracy: 0.9991 - val_loss: 0.0203 - val_accuracy:
0.9982 - lr: 3.6788e-05
Epoch 21/35
35/35 [==============================] - 29s 831ms/step - loss: 0.0083 - accuracy: 0.9998 - val_loss: 0.0179 - val_accuracy:
0.9982 - lr: 3.4994e-05
Epoch 22/35
35/35 [==============================] - 29s 835ms/step - loss: 0.0077 - accuracy: 1.0000 - val_loss: 0.0157 - val_accuracy:
0.9982 - lr: 3.3287e-05
Epoch 23/35
35/35 [==============================] - 28s 807ms/step - loss: 0.0076 - accuracy: 1.0000 - val_loss: 0.0132 - val_accuracy:
0.9982 - lr: 3.1664e-05
Epoch 24/35
35/35 [==============================] - 29s 827ms/step - loss: 0.0073 - accuracy: 0.9998 - val_loss: 0.0136 - val_accuracy:
0.9982 - lr: 3.0119e-05
Epoch 25/35
35/35 [==============================] - 30s 855ms/step - loss: 0.0076 - accuracy: 0.9995 - val_loss: 0.0124 - val_accuracy:
0.9982 - lr: 2.8651e-05
Epoch 26/35
35/35 [==============================] - 31s 897ms/step - loss: 0.0066 - accuracy: 0.9998 - val_loss: 0.0117 - val_accuracy:
0.9982 - lr: 2.7253e-05
Epoch 27/35
35/35 [==============================] - 28s 802ms/step - loss: 0.0065 - accuracy: 0.9998 - val_loss: 0.0117 - val_accuracy:
0.9982 - lr: 2.5924e-05
Epoch 28/35
35/35 [==============================] - 30s 847ms/step - loss: 0.0061 - accuracy: 0.9998 - val_loss: 0.0108 - val_accuracy:
0.9982 - lr: 2.4660e-05
Epoch 29/35
35/35 [==============================] - 30s 848ms/step - loss: 0.0062 - accuracy: 1.0000 - val_loss: 0.0116 - val_accuracy:
0.9982 - lr: 2.3457e-05
Epoch 30/35
35/35 [==============================] - 29s 834ms/step - loss: 0.0058 - accuracy: 1.0000 - val_loss: 0.0109 - val_accuracy:
0.9982 - lr: 2.2313e-05
Epoch 31/35
35/35 [==============================] - 30s 862ms/step - loss: 0.0060 - accuracy: 1.0000 - val_loss: 0.0110 - val_accuracy:
0.9982 - lr: 2.1225e-05
Epoch 32/35
35/35 [==============================] - 30s 844ms/step - loss: 0.0054 - accuracy: 1.0000 - val_loss: 0.0108 - val_accuracy:
0.9982 - lr: 2.0190e-05
Epoch 33/35
35/35 [==============================] - 30s 857ms/step - loss: 0.0056 - accuracy: 1.0000 - val_loss: 0.0105 - val_accuracy:
0.9982 - lr: 1.9205e-05
Epoch 34/35
35/35 [==============================] - 29s 827ms/step - loss: 0.0053 - accuracy: 1.0000 - val_loss: 0.0098 - val_accuracy:
0.9982 - lr: 1.8268e-05
Epoch 35/35
35/35 [==============================] - 30s 862ms/step - loss: 0.0051 - accuracy: 1.0000 - val_loss: 0.0097 - val_accuracy:
0.9982 - lr: 1.7377e-05
18/18 [==============================] - 1s 67ms/step - loss: 0.0094 - accuracy: 0.9982
Test Accuracy: 0.9982
Test Loss: 0.0094
```

Figure 5: Training the Model on the 'CRMFR Dataset'